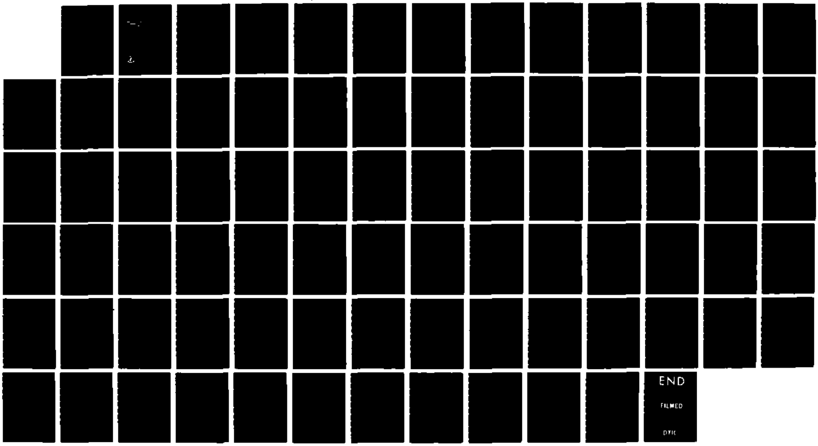
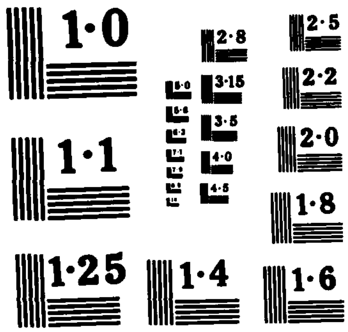
AD-A157 682 SHIPFIT - A COMPUTER PROGRAM FOR DETERMINING THE ADDED MASSES OF SHIPS AND SUBMERSIBLES(U) NAVAL SURFACE WEAPONS CENTER SILVER SPRING MD W W MCDONALD 07 JUN 84 1/1

UNCLASSIFIED NSWC/TR-83-460 F/G 9/2 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

(2)

NSWC TR 83-460

AD-A157 682

SHIPFIT -- A COMPUTER PROGRAM FOR DETERMINING THE ADDED MASSES OF SHIPS AND SUBMERSIBLES

BY WILLIAM W. McDONALD

RESEARCH AND TECHNOLOGY DEPARTMENT

7 JUNE 1984

NAVAL SURFACE WEAPONS CENTER

Dahlgren, Virginia 22448 • Silver Spring, Maryland 20910

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NSWC TR 83-460	2. GOVT ACCESSION NO. AD-A157682	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SHIPFIT--A COMPUTER PROGRAM FOR DETERMINING THE ADDED MASSES OF SHIPS AND SUBMERSIBLES		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William W. McDonald		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Surface Weapons Center (Code R14) White Oak Silver Spring, MD 20910		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62633N; F33327; SF33327; R19BA
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 7 June 1984
		13. NUMBER OF PAGES 81
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program	Strip Theory	Vibrational Motion
Added Mass	Potential Flow	Mass distribution
Virtual Mass	Spline Function	Flooding
Ship Sections	Conformal Mapping	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SHIPFIT, a computer program that calculates the added mass per unit length of ship or submersible, is described. A conformal mapping technique is used that is widely employed in problems permitting a strip theory approach. The scheme differs from other published methods in that the ship form is initially fitted to a cubic spline function. The coefficients of the conformal transformation are then obtained from the cubic spline function by a least squares iteration technique.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

The computer program SHIPFIT was originally published in an internal NSWC Technical Note dated 23 April 1976. Since that time SHIPFIT has been used to determine the added masses of ships and submersibles in a number of ship response studies conducted by the Explosion Damage Branch (R14) of this Center. Recent requirements for horizontal added masses and the desire to reference the computer program in publications distributed externally have brought about this updating of the program and method.

This revision was sponsored by NAVSEA (63R) through the Undersea Weapons, Warheads, and Fuzes Block program.

Approved by:

KURT F. MUELLER, Acting Head
Energetic Materials Division

CONTENTS

ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

The approach taken in this report follows the conventional strip-theory technique for estimating the distribution of added mass along the length of a ship undergoing transverse vibrational motions. The flow is considered to be inviscid, incompressible, and irrotational, and, within cross-sectional planes spaced length-wise along the hull, to be two-dimensional. The portion of the ship between two planes is referred to as a strip or segment, and the intersection of a plane with the hull is called a section. In strip theory ship sections are considered to be unvarying in form within each segment.

The added mass of a ship segment in this two-dimensional flow field is usually a close approximation to the added mass under more realistic conditions of flow and can be calculated using complex variables theory and conformal mapping techniques. Once obtained, this approximation may be improved by multiplying it by a reduction factor equal to the ratio of the approximate two-dimensional and exact three-dimensional added masses of a vibrating ellipsoid of revolution fitted to the ship for which an exact solution is known. The added masses calculated by SHIPFIT do not contain the reduction factor.

The theory employed in SHIPFIT is adapted from the 1957 paper of Landweber and Macagno,[1] in which formulas were developed for the added masses of floating and submerged bodies, of arbitrary cross section, oscillating in both horizontal and vertical directions. The added masses are expressed in terms of the coefficients of a conformal transformation that maps the ship cross-sectional form onto a circle of fixed radius. In SHIPFIT, the forms of these expressions are slightly changed to facilitate their coding. For totally submerged bodies, the theory is somewhat modified so that the cross sections of submersibles can be viewed in their usual upright orientation and to facilitate the evaluation of the mapping function coefficients.

The method by which the coefficients of the conformal transformation are obtained differs from previously published techniques. In SHIPFIT, a cubic spline function is initially fitted to the data points describing the section form. The mapping function is then fitted to the spline function by means of a least squares Newton-Raphson technique. Benefits of the method appear to include a considerable improvement in both speed and accuracy over other published methods.*

In addition to calculating the added section masses per unit length, SHIPFIT includes, as an option, a provision for determining the equilibrium orientation of the ship with or without user-specified conditions of compartment flooding. This is done by the subroutine ORIENT prior to the added mass calculations.

*This paragraph refers to the state-of-the-art in 1976 when SHIPFIT was first developed (see the Foreword).

CHAPTER 2

THEORY AND METHOD

The theory applied to the case of a ship (or submersible) floating on the surface is identical to that of Landweber and Macagno[1] with the exception of some changes in notation. Therefore, only results and a brief description for this case will be included. The theory employed in SHIPFIT for a submerged vessel, however, differs somewhat from that of Reference 1 and, consequently, will be developed more fully.

The use of conformal mapping to obtain the added masses of objects in two-dimensional potential flows is a common topic in hydrodynamics texts. Basically it consists of finding a transformation that will map the streamlines of the object onto those of a circle or flat plane where the complex potential and kinetic energy integral are more easily derived. The added mass is then readily obtained.

In Reference 1, floating and submerged hulls are handled in very similar ways. The ship on the surface is actually treated as if it were a completely submerged "double hull" formed by reflecting the submerged portion of the hull in the surface. The two cases of interest, then, are illustrated in Figure 1. The two halves of the "double hull" are imagined to vibrate in such a way that the free surface boundary condition is satisfied along the reflection line exterior to the ship form. The appropriate boundary condition for most cases of interest is that the velocity potential vanish along the free surface (this is actually the high frequency condition). For vertical motion this boundary condition is satisfied if both halves of the double hull vibrate together as a single rigid form. For horizontal motion, the vibrations of the two halves are equal but in opposite directions. In both cases, the added mass of the double hull is twice that of the desired value.

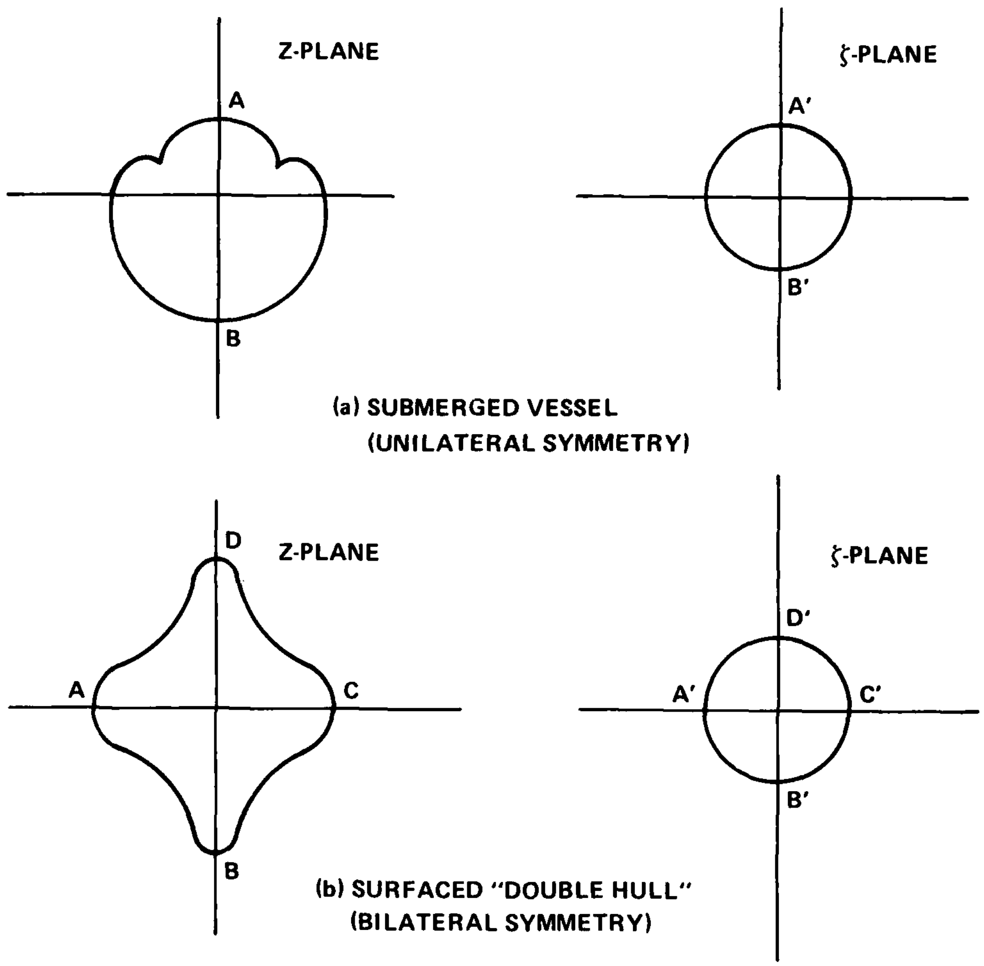


FIGURE 1. CASES OF INTEREST

For the two cases illustrated in Figure 1, Landweber and Macagno map the streamlines of the (single or double) ship form onto those exterior to a circle of radius r_o by using the transformation

$$Z = \zeta + \frac{a_1'}{\zeta} + \frac{a_2'}{\zeta^2} + \ldots , \tag{1}$$

where Z is a point in the complex plane containing the ship section, hereafter called the Z-plane, and $\zeta = re^{i\theta}$ is a point in the ζ-plane containing the circle of radius r_o. The representation of the complex potential $\varphi + i\psi$ in the ζ-plane is assumed to be of the form

$$\varphi + i\psi = \frac{b_1'}{\zeta} + \frac{b_2'}{\zeta^2} + \ldots , \tag{2}$$

where φ and ψ are, of course, the velocity potential and stream function. By substitution of the representations of φ and ψ from Equation (2) into the kinetic energy integral of a fluid at rest at infinity, i.e.,

$$T = -1/2\ \rho \oint \phi d\psi , \tag{3}$$

where ρ is the fluid density and the integration is over all boundaries of the fluid, Landweber and Macagno show that the kinetic energy and added mass per unit length can be expressed as rather simple sums involving the coefficients b_1', b_2', ... Finally, the b' coefficients are expressed in terms of the a' coefficients, which map the ship contour onto the circle, by satisfying the boundary condition that requires the normal velocity of the fluid to equal that of the ship at the ship contour.

The transformation function given as Equation (1) is quite general and may be used to represent an arbitrary shape. It is desirable, however, to build the symmetry properties exhibited by the forms of Figure 1 into the mapping function. It is easily verified that the following relationships must be satisfied:

UNILATERAL SYMMETRY (about vertical axis)
$$Z(-\bar{\zeta}) = -\bar{Z}(\zeta), \tag{4}$$

BILATERAL SYMMETRY
$$Z(\bar{\zeta}) = \bar{Z}(\zeta) . \tag{5}$$

Here the overbar denotes the complex conjugate. By substitution of Equation (1) into (4), one finds that unilateral symmetry requires that the coefficients of odd powers of ζ be purely real and the coefficients of even powers of ζ be purely imaginary. In the same manner, bilateral symmetry may be shown to require real odd coefficients and vanishing even coefficients.

Added Masses of a Ship Floating on the Surface

For the bilaterally symmetric case it is convenient to express the mapping function as

$$Z = \sum_{n=1}^{\infty} a_n \zeta^{3-2n} , \tag{6}$$

where, as stated above, the coefficients are now real. This is the form actually used in SHIPFIT. Also in SHIPFIT, the ship form is mapped onto the unit circle ($r_0=1$) rather than onto a circle of arbitrary radius. The differences between Equation (6) and Equation (1) (applied to the case of bilateral symmetry) are purely notational. The correspondence between coefficients is easily shown to be

$$a_1 = r_o; \quad a_n = a'_{2n-3} / r_o^{2n-3} \quad , n > 1 \ . \tag{7}$$

For the bilaterally symmetric case, the expressions for added mass derived by Landweber and Macagno can now be written in the form that appears in SHIPFIT by substituting Equations(7). The added mass per unit length for vertical oscillations becomes

$$A_V = \pi\rho \ a_1(a_1 + a_2) - \rho S \ , \tag{8}$$

where

$$S = \frac{\pi}{2} \sum_{n=1}^{\infty} (3-2n) \ a_n^2 \ . \tag{9}$$

S is shown by Landweber and Macagno to be simply the submerged area of the ship cross section (half the area of the "double hull"). This result is obtained by integrating $1/2 \oint xdy$ over the double hull contour after substituting parametric equations for x and y in terms of θ.

In the horizontal direction the result given by Landweber and Macagno may be written as

$$A_H = \frac{8\rho}{\pi} \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \gamma_{jk} \ \bar{a}_{j+1} \ \bar{a}_{k+1} \ , \tag{10}$$

CHAPTER 3

USE OF THE PROGRAM

Input data is supplied to SHIPFIT through three separate files: DATA1, DATA2, and INPUT. DATA1 and DATA2 are presently prepared using an electronic digitizer from drawings of the ship cross sections and the ship weight histogram, respectively. The weight distribution is needed only for surface ships or surfaced submersibles and only if a computed equilibrium position is desired.

DATA1 File

File DATA1 contains the x and y coordinates relative to the data origin of points lying along the ship forms and the locations of the calculation origins. Ship forms are assumed to be symmetric and either the left or right half of a particular cross section may be used. The order of the points on DATA1 is important; an improper ordering will stop the program execution and cause the message "IMPROPER DATA FORMAT OR CALCULATION ORIGIN FOR SECTION __ ... STOP" to be printed. The first point of each section is the calculation origin (see definition, page 14). The second and successive points up to a maximum of 64 must begin in the vicinity of the lowest point of intersection of the ship hull with the vertical symmetry axis and continue in steps along the hull in either a positive or negative sense of rotation about the calculation origin. For surface ships, the last point should define the freeboard of the section. For submersibles, the last point should be in the vicinity of the uppermost point of intersection of the hull with the vertical axis. The data are read in the main subprogram of SHIPFIT by the list directed read statement

```
READ (1,*) IDIG, XDIG, YDIG
```

Determination of the Ship Equilibrium Position

As an option, SHIPFIT determines the orientation of a ship floating on the surface. This is done by satisfying the two equations

$$\sum_i B_i = \sum_j W_j \tag{44}$$

$$\sum_i B_i z_{Bi} = \sum_j W_j z_{Wj} \ , \tag{45}$$

where B_i is the lumped buoyancy force of the ith ship segment centered at z_{Bi} and W_j is the lumped weight from the jth column of the ship weight histogram centered at z_{Wj}. Both distances are measured from the forward perpendicular (forward-most end of the designer's waterline). A provision for additional weights, due, for example, to flooding is included. Ship segments pertaining to missing cross section data are considered neutral. Equations (44) and (45) are satisfied iteratively in subroutine ORIENT by adjusting the equation of the ship waterline.

For sections with bilateral symmetry, all but the first two coefficients are assigned inital values of zero. Initial values for a_1 and a_2 are then obtained from the half beam h at the waterline and draft d of the section. Using Equations (33), (34), (37), and (38) and setting values of θ to 0 and $-\pi/2$, we obtain

$$a_1 = (d + h)/2$$

$$a_2 = (d - h)/2 \tag{42}$$

The initial coefficients for sections with unilateral symmetry are obtained from the form limits along the y axis, say y_s^+ and y_s^-, and the enclosed area A. A is calculated from the spline function fit to the section. All other coefficients are zero. The values used are

$$a_1 = \frac{(y_s^+ - y_s^-)}{4} + \frac{A}{\pi(y_s^+ - y_s^-)}$$

$$a_2 = (y_s^+ + y_s^-)/2 \tag{43}$$

$$a_3 = a_1 - (y_s^+ - y_s^-)/2$$

This set of initial values actually amounts to a four parameter fit to the form as a_4 is then identically zero. The coefficients specify an ellipse enclosing an area A and passing through the end points y_s^+ and y_s^-.

are sought. $R(\phi_c)$ is a functional representation of the ship hull relative to the calculation origin. SHIPFIT uses a cubic spline function for this purpose because of its desirable smoothing properties, speed of calculation, and the continuity of its first and second derivatives.

The Newton-Raphson technique is an iteration scheme in which the $\partial\Sigma/\partial a_n$ are expanded up to first order in Taylor series about the initial set of coefficients. We obtain

$$\frac{\partial\Sigma}{\partial a_n}(\underline{a}) \approx \frac{\partial\Sigma}{\partial a_n}(\underline{a}^o) + \sum_{k=1}^{N} \frac{\partial^2\Sigma}{\partial a_k \partial a_n}(\underline{a}^o)\,[a_k - a_k^o], \quad n=1,2,\ldots N, \tag{40}$$

where the underbar denotes the full vector of coefficients. The right hand sides of Equation (40) are equated to zero and the set of N linear equations are solved for a new set of coefficients $\{a_k\}$, which is used to replace the old in the next iteration step.

SHIPFIT employs a technique to accelerate this process that is found in the NSWC/WOL Library subroutine LSQSUB. To appreciate this feature let us first expand the second derivative appearing in Equation (40).

$$\frac{\partial^2\Sigma}{\partial a_k \partial a_n} = 2 \sum_{m=1}^{M} \left[\frac{\partial\Delta_m}{\partial a_k}\frac{\partial\Delta_m}{\partial a_n} + \Delta_m \frac{\partial^2\Delta_m}{\partial a_k \partial a_n}\right] \tag{41}$$

where $\Delta_m \equiv r_{cm} - R(\phi_{cm})$. The technique consists of deleting the second derivative term on the right hand side and approximating $\partial^2\Sigma/\partial a_k \partial a_n$ by the remaining terms (twice the sums of the products of the first derivatives). This reduces the computation time significantly as second derivatives of Δ_m need not be calculated. If the process fails to produce convergence in a given number of iterations, SHIPFIT repeats the process with the exact expression for the second derivative.

For an approximately correct initial set of coefficients $\{a_n^o\}$ and unilateral symmetry, a half plane in the ζ-plane will map onto the same half plane in the Z-plane. Similarly, for the bilaterally symmetric transformation, quadrants will map onto quadrants. Such mappings are suggested in Figure 1 by the primed and unprimed sets of points and image points A,B,C,D and A',B',C',D'. SHIPFIT creates an array of points, $e^{i\theta_m}$, m = 1,2, ... M, evenly spaced on the unit circle and spanning the quadrant or the half plane of interest. Images $\{x_{sm}, y_{sm}\}$ of these points, then, lie more or less near the ship form in the Z-plane depending upon the correctness of the initial choices of the coefficients. The problem consists of finding a way of improving the initial guesses for the coefficients so that the points $\{x_{sm}, y_{sm}\}$ lie on or very close to the ship form. These points will also span the quadrant or half plane of interest in the Z-plane, but the spacing between the points in the Z-plane will in general not be uniform.

It should be noted that by fixing the points in the ζ-plane, Equations (33) and (34) become very simple. As noted above, the functions $c_x(m,n)$ and $c_y(m,n)$ become fixed constants, and for a reasonable range of n values, they may be tabulated prior to the task of improving the coefficient values. In the paper by von Kerczek and Tuck[2], points in the Z-plane were held fixed and their corresponding image points in the ζ-plane were left free. The location of the ζ-plane points, then, required the solution by iteration of M transcendental equations involving the trigonometric functions. Computation times using their method are thus considerably greater than those for the present method.

Adjustment of the coefficients to increase the closeness of the fit is accomplished in SHIPFIT by means of a least squares technique. The points $\{x_{sm}, y_{sm}\}$ are first transformed to a polar representation, $\{r_{cm}, \phi_{cm}\}$, about the calculation origin. Then, by means of a Newton-Raphson technique, the set of coefficients satisfying the equations

$$\frac{\partial \Sigma}{\partial a_n} \equiv \frac{\partial}{\partial a_n} \left[\sum_{m=1}^{M} (r_{cm} - R(\phi_{cm}))^2\right] = 0, \quad n=1,2,\ldots N \tag{39}$$

$$x_{sm} = \sum_{n=1}^{\infty} c_x(m,n)\ a_n \tag{33}$$

$$y_{sm} = \sum_{n=1}^{\infty} c_y(m,n)\ a_n\ . \tag{34}$$

In the case of unilateral symmetry, the quantities c_x and c_y are identified from Equations (24) and (25) as

$$c_x(m,n) = \begin{cases} \cos\ [(2-n)\theta_m],\ n\ \text{odd} \\ -\sin\ [(2-n)\theta_m],\ n\ \text{even} \end{cases} \tag{35}$$

$$c_y(m,n) = \begin{cases} \sin\ [(2-n)\theta_m],\ n\ \text{odd} \\ \cos\ [(2-n)\theta_m],\ n\ \text{even}\ . \end{cases} \tag{36}$$

In the bilaterally symmetric case, the c_x and c_y quantities are more simply expressed as

$$c_x(m,n) = \cos\ [(3-2n)\theta_m] \tag{37}$$

$$c_y(m,n) = \sin[(3-2n)\theta_m]\ . \tag{38}$$

Hence, for a fixed set of points on the unit circle in the ζ-plane corresponding to a fixed set of angles, the quantities c_x and c_y in Equations (33) and (34) are also fixed.

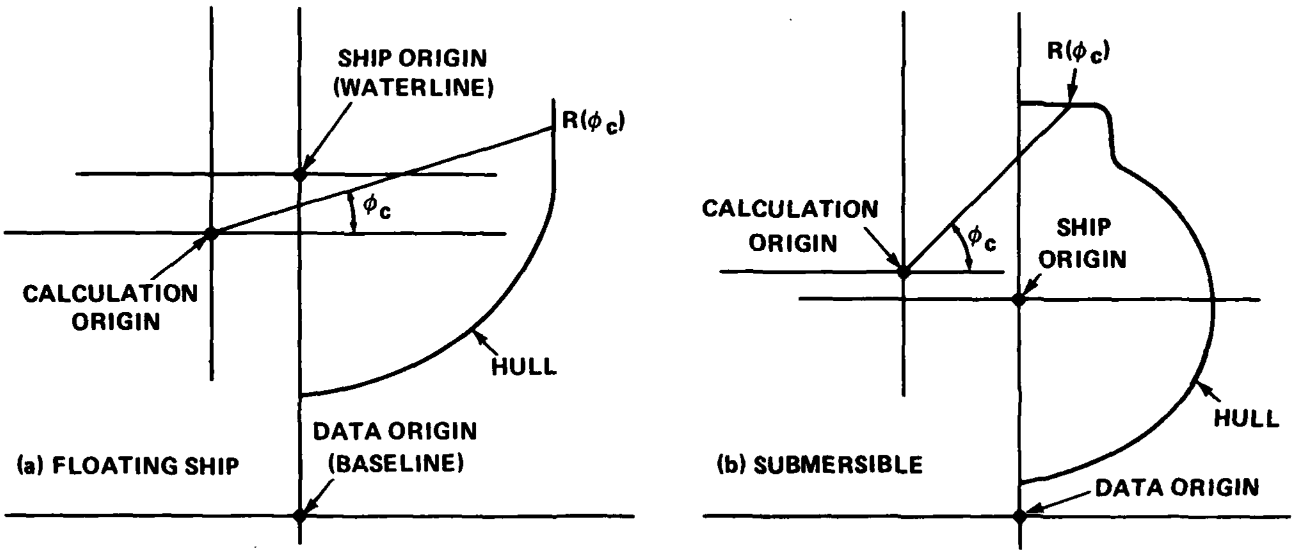


FIGURE 2. COORDINATE SYSTEMS USED IN <u>SHIPFIT</u>

Fitting of the Mapping Function Coefficients

An understanding of the program and the method used for obtaining the coefficients requires the definitions of the three coordinate systems employed. These are shown in Figures 2a and 2b. As they are related through simple translations, they are described by definitions of their origins.

The Data Origin: This is the reference origin for the (x,y) coordinates of input points lying along the section that define its shape. The data origin must always be along the vertical axis, and for ships floating on the surface, it must be the same for all sections. A convenient location is the intersection of the vertical symmetry axis with the base line.

The Ship Origin: This is the origin of the Z plane. For surface ships, it is taken as the intersection of the vertical axis with the water line. For submerged vessels, it is located at the geometric center of the form.

The Calculation Origin: This is a somewhat arbitrary point chosen so that, in a polar representation about this origin, the radius of the ship form can be described by a single valued function of the angle. This is a relatively minor constraint on the generality and utility of SHIPFIT as a calculation origin can almost always be found.

In the text that follows, the subscripts d, c, and s will be used to reference variables to a particular coordinate system (<u>d</u>ata, <u>c</u>alculation, or <u>s</u>hip).

The objective of the fitting process is to find the set of coefficients $\{a_n\}$, that will make points lying on the unit circle in the ζ-plane, say $\{e^{i\theta_m}$, $m=1,2,\ldots, M\}$, the images under the mapping transformation of points lying on the submerged portion of the ship hull in the Z-plane, say $\{Z_{sm}$, $m=1,2\ldots, M\}$ (reference is to the ship origin).

For either a unilaterally or bilaterally symmetric ship form, the x and y coordinates of the points in the Z-plane can be expressed generally as linear combinations of the unknown parameters $\{a_n\}$, i.e.,

$$b_j = \begin{cases} iUa_{j+2}, & j \geq 2 \text{ and even} \\ Ua_{j+2}, & j \geq 3 \text{ and odd} . \end{cases} \tag{29}$$

These produce an added mass per unit length for a submerged vessel vibrating horizontally of the form

$$A_H = 2\pi\rho\ a_1(a_1 - a_3) - \rho S' , \tag{30}$$

where S' is again given by Equation (28).

Comparison of the formulas for A_V and A_H with those obtained by Landweber and Macagno shows the forms of the expressions to be identical, even though the symmetry of the cross section is now about the vertical axis rather than the horizontal or imaginary axis. It should be remembered, however, that the coefficients of the mapping function differ in the two approaches.

Added mass coefficients for the case of a submerged vessel are defined as

$$C_V = \frac{A_V}{\pi\rho h^2} \tag{31}$$

and

$$C_H = \frac{A_H}{\pi\rho(H/2)^2} , \tag{32}$$

where h is the half beam of the submerged form and H is the full height from top to bottom of the section.

The b coefficients for vertical oscillations are found by inserting Equations (18) and (24), with r=1, into Equation (22) and matching the coefficients of the two power series. We obtain

$$b_0 \text{ real}, \ b_1 = -iV(a_1 + a_3) \ ,$$

$$b_j = \begin{cases} Va_{j+2} \ , & j \geq 2 \text{ and even} \\ -iVa_{j+2} \ , & j \geq 3 \text{ and odd} \ . \end{cases} \tag{26}$$

Substituting these values into Equation (20), we obtain the added mass per unit length of a submerged vessel vibrating vertically,

$$A_V = 2\pi\rho \ a_1(a_1 + a_3) - \rho S', \tag{27}$$

where

$$S' = \pi \sum_{n=1}^{\infty} (2-n) \ a_n^2 \tag{28}$$

is the cross-sectional area of the submersible. The area result follows upon substitution of Equations (24) and (25) into $\oint x(\partial y/\partial\theta)d\theta$.

In a similar manner, b coefficients for oscillations in the horizontal direction are found to be

$$b_0 = iUa_2 \ , \ b_1 = U(a_3 - a_1) \ ,$$

$$A_H = \frac{\pi\rho}{U^2} \sum_{n=1}^{\infty} n \left|b_n\right|^2 . \tag{21}$$

These results are obtained by simply equating T from Equation (19) and $1/2\ A_V V^2$ or $1/2\ A_H U^2$.

The b coefficients in Equations (20) and (21) may be expressed in terms of the a coefficients of Equation (15) by satisfying the boundary conditions on the ship contour. Landweber and Macagno show these to be

$$\psi = -Vx \tag{22}$$

for vertical oscillations (velocity V) and

$$\psi = Uy \tag{23}$$

for horizontal oscillations (velocity U). Here, x and y are the real and imaginary parts of Z on the ship contour, which may be expressed as

$$x = \frac{1}{2} \sum_{n\ \mathrm{odd}} a_n \left(e^{i(2-n)\theta} + e^{-i(2-n)\theta}\right)$$

$$- \frac{1}{2i} \sum_{n\ \mathrm{even}} a_n \left(e^{i(2-n)\theta} - e^{-i(2-n)\theta}\right), \tag{24}$$

$$y = \frac{1}{2} \sum_{n\ \mathrm{even}} a_n \left(e^{i(2-n)\theta} + e^{-i(2-n)\theta}\right)$$

$$+ \frac{1}{2i} \sum_{n\ \mathrm{odd}} a_n \left(e^{i(2-n)\theta} - e^{-i(2-n)\theta}\right) . \tag{25}$$

where b_0 is a complex constant. Substituting $\zeta = re^{i\theta}$, we may express the real and imaginary parts of this as

$$\varphi = \frac{1}{2} \sum_{n=0}^{\infty} \frac{1}{r^n} (b_n e^{-in\theta} + \bar{b}_n e^{in\theta}) \tag{17}$$

$$\psi = \frac{1}{2i} \sum_{n=0}^{\infty} \frac{1}{r^n} (b_n e^{-in\theta} - \bar{b}_n e^{in\theta}) . \tag{18}$$

It is easily shown* that this form of the complex potential yields the same result for the kinetic energy as obtained by Landweber and Macagno, i.e.,

$$T = \frac{\pi\rho}{2} \sum_{n=1}^{\infty} n \left|b_n\right|^2 . \tag{19}$$

This is obtained by inserting Equations (17) and (18) into Equation (3).

The added masses per unit length for submerged hulls vibrating vertically with velocity V or horizontally with velocity U are, respectively,

$$A_V = \frac{\pi\rho}{V^2} \sum_{n=1}^{\infty} n \left|b_n\right|^2 \tag{20}$$

*Proof is as follows. Let $\varphi' = \varphi + \varphi_0$ and $\psi' = \psi + \psi_0$. Then

$$\oint \varphi' d\psi' = \oint \varphi d\psi + \varphi_0 \oint d\psi .$$

But $\oint d\psi = 0$ since the integration may proceed around the ship form, to infinity along a streamline, around the circle at infinity and back to the ship.

Added Masses of a Submerged Vessel

For the case of unilateral symmetry (i.e., a submersible operating at a depth where surface effects are negligible), it is convenient to use a mapping function of the form

$$Z = \sum_{\substack{n=1 \\ n \text{ odd}}}^{\infty} a_n \zeta^{2-n} + i \sum_{\substack{n=2 \\ n \text{ even}}}^{\infty} a_n \zeta^{2-n} , \tag{15}$$

where the coefficients are all real. This form differs from that used by Landweber and Macagno in two respects. First, Equation (1) does not include an additive constant while Equation (15) does (a_2). This is useful in the fitting process. Second, for real coefficients, Equation (1) has unilateral symmetry about the real axis rather than about the imaginary axis. Thus, to apply the results of Landweber and Macagno to submersibles, the cross sections must be rotated 90 degrees. It is desirable to have a theory where such a rotation is unnecessary.

Because velocities are invariant under changes of the complex potential by an additive constant, the transformation for the complex potential, Equation (2), can be written more generally as

$$\varphi + i\psi = b_0 + \frac{b_1}{\zeta} + \frac{b_2}{\zeta^2} + \dots , \tag{16}$$

where

$$\bar{a}_2 \equiv a_2 - a_1 \; ; \quad \bar{a}_j \equiv a_j, \; j > 2 \quad , \tag{11}$$

and the coefficients γ_{jk}, $j,k \geq 1$ are computed by the following expressions:

$$\gamma_{jk} = \gamma_{kj} = \begin{cases} -\dfrac{(2j-1)(2k-1)}{4(k-j)} \displaystyle\sum_{m=1}^{k-j} \dfrac{1}{[2(j+m)-1][2(k-m)-1]} \; , \; j<k \\ \\ \dfrac{(2j-1)}{4} \displaystyle\sum_{m=1}^{2j-1} \dfrac{1}{[2(j-m)-1]^2} \; , \; j=k \; . \end{cases} \tag{12}$$

Added mass coefficients are defined as the ratio of the added mass of the ship form divided by the added mass of a similar, circular form. For a ship floating on the surface, these are given as

$$C_V = \frac{A_V}{\pi \rho h^2/2} \tag{13}$$

and

$$C_H = \frac{A_H}{\pi \rho d^2/2} \; , \tag{14}$$

where h is half beam of the cross section at the waterline and d is the ship draft.

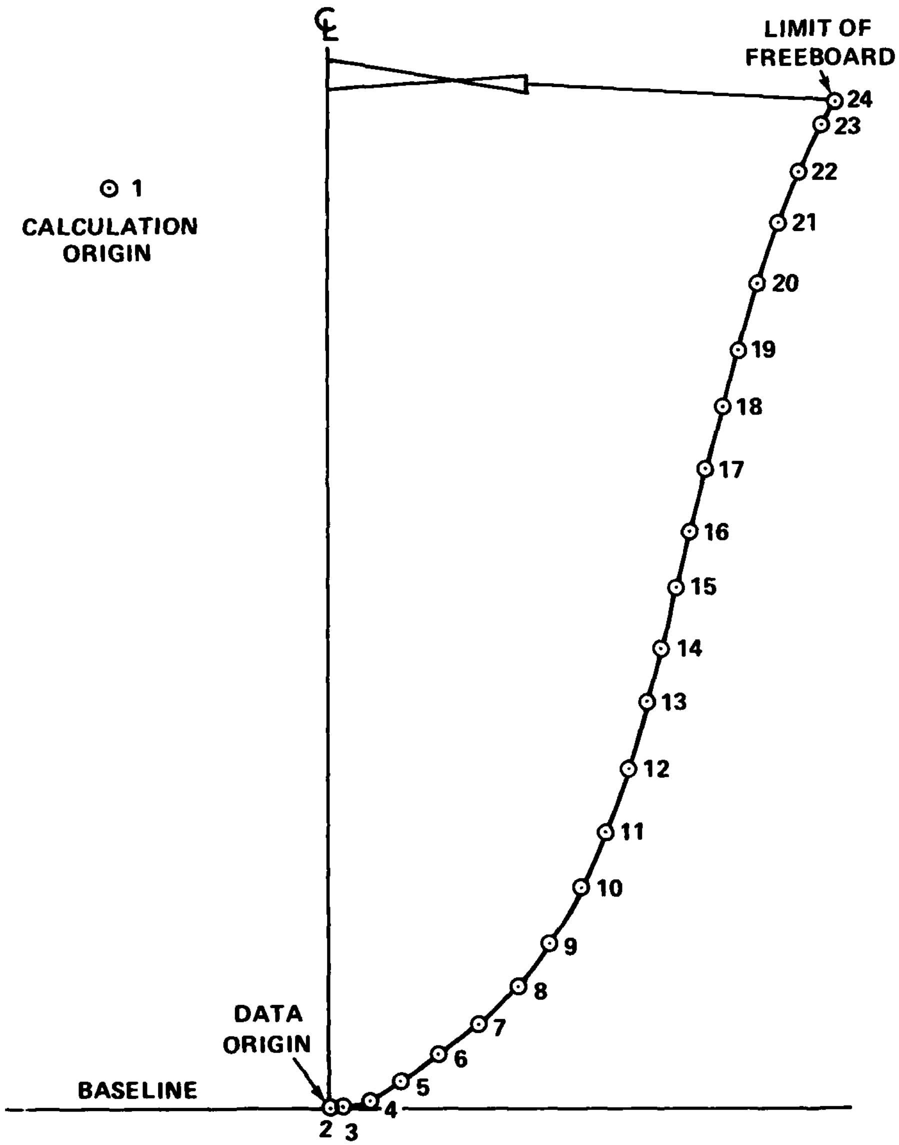
where IDIG is the sequential integer count of points pertaining to a particular cross section. Each time data for a new cross section appears in the file (i.e., a calculation origin) the value of IDIG must be reset to one. Data for a total of 24 cross sections may appear on DATA1. Finally, points should be more closely spaced where the ship form makes sharp turns or has a small radius of curvature. This is because the derivatives of the spline function fitted to the form are continuous. An example of DATA1 input for a surface ship appears in Figure 3. Note that the data origin is set at the intersection of the centerline and baseline, and that the calculation origin is arbitrarily located, but chosen so that $R(\phi)$ is not highly varying. Point coordinates for each section may be multiplied by an arbitrary scale factor (see INPUT file instructions).

DATA2 File

File DATA2 contains the ship weight distribution. It will be used if the ship is floating on the surface and if the equilibrium calculation is selected by setting EQCAL to TRUE on the INPUT file; otherwise, DATA2 is not required. Data are read in the subprogram ORIENT by the list directed read statement

```
READ (2,*) IDIG, ZDIG, WZDIG
```

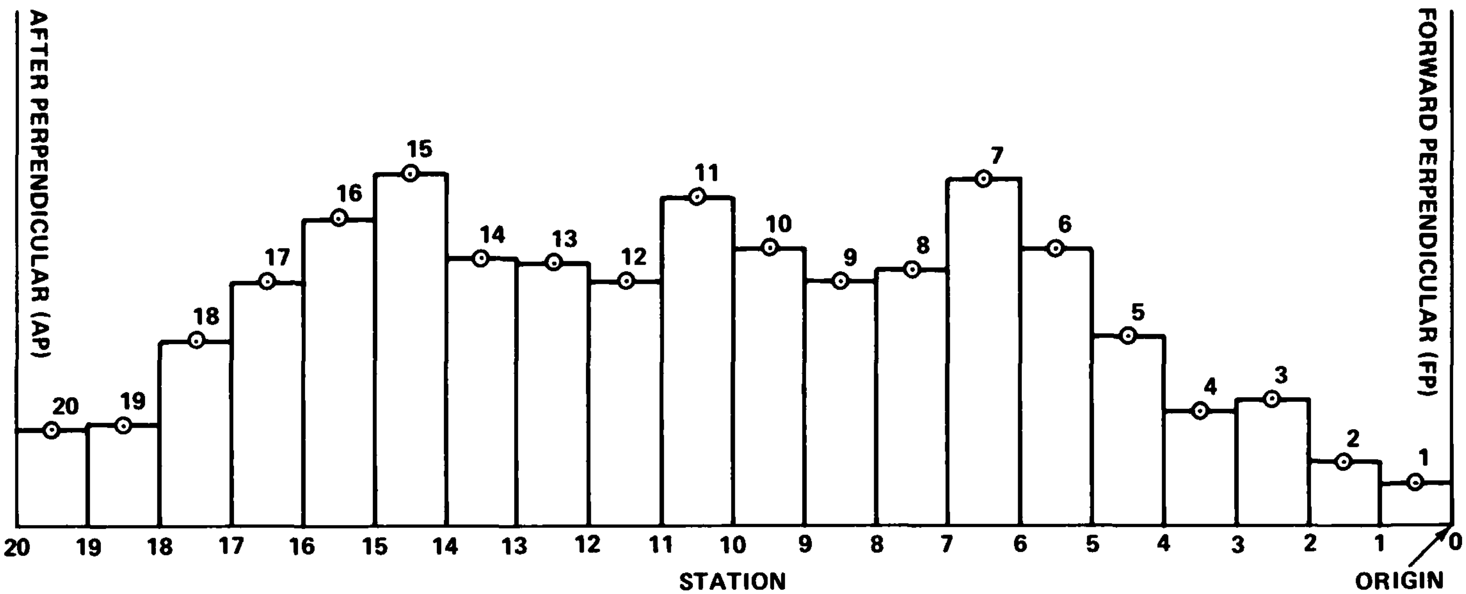
where IDIG is a point counter (for convenience only), ZDIG is the position relative to the forward perpendicular of the center of a weight histogram column, and WZDIG is the column height, giving weight per unit length. The ZDIG scale is defined as zero at the forward perpendicular (bow) and positive in the direction of the after perpendicular (stern). The ordering of points on DATA2 is not important. Any point sequence may be used, but all weight histogram columns should be represented, up to a maximum of 48 columns (the program only uses those columns that overlap ship segments included on the DATA1 file). Arbitrary scale multipliers may be read in from the INPUT file. An example of DATA2 input appears in Figure 4. (Note: the number of weight histogram columns does not need to equal the number of ship segments.)



DATA FILE:

	IDIG	XDIG	YDIG
THIS SECTION	1	-6.157	26.00
	2	0	0.
	3	.3733	.4930E-02
	4	1.118	.1748
	5	1.963	.7726
	6	3.073	1.534
	7	4.235	2.403
	8	5.342	3.431
	9	6.179	4.615
	10	7.064	6.280
	11	7.844	7.837
	12	8.514	9.606
	13	9.048	11.60
	14	9.428	13.03
	15	9.832	14.80
	16	10.24	16.35
	17	10.69	18.17
	18	11.20	19.94
	19	11.61	21.59
	20	12.12	23.41
	21	12.73	25.18
	22	13.41	26.58
	23	13.98	27.92
	24	14.34	28.56
NEXT SECTION	1	-6.124	21.54
	2	-.5330E-01	.1748E-02
	3	.3731	-.1223E-01
	4	1.288	.2246
	5	2.588	.8223
	6	4.325	1.726
	7	6.288	2.995
	8	7.995	4.594
	9	9.221	6.208
	10	10.63	8.619
	⋮		

FIGURE 3. EXAMPLE OF DATA1 FILE LISTING FOR A SURFACE SHIP CROSS SECTION



IDIG	ZDIG	WZDIG	IDIG	ZDIG	WZDIG
1	9.695	1.740	11	200.6	13.66
2	28.84	2.512	12	219.6	10.11
3	47.52	5.185	13	238.4	10.94
4	67.11	4.747	14	257.2	11.08
5	86.14	7.881	15	276.5	14.62
6	105.1	11.48	16	295.8	12.85
7	124.6	14.32	17	315.1	10.11
8	143.6	10.71	18	334.4	7.768
9	162.4	10.16	19	354.2	4.218
10	181.5	11.57	20	373.4	4.010

FIGURE 4. EXAMPLE OF DATA2 FILE FOR A PARTICULAR WEIGHT HISTOGRAM

INPUT File

Data on the INPUT file control the operation of the program and provide the values of certain necessary quantities. The structure of the data on INPUT will vary according to the intentions of the user. A number of INPUT records are conditional upon the values of variables read previously, as indicated in the READ statement structure appearing in Figure 5. Listed below are variable definitions in the order in which they are read. Line designations match those in Figure 5. Many variables are provided with default values that are used if the corresponding input field is left blank. These are underlined in Figure 5. A list of these variables along with their default values appears in Figure 6. Although physical dimensions are stated, any standard consistent system may be used (Caution: see SPVOL, Line b).

INPUT File Variable Definitions (Refer to lines in Figure 5)

Line a (Title):

TITLE is an arbitrary character array that is printed on each page of output and each plot.

Line b (Main Program Control Parameters):

SUBMERG is TRUE if the ship is submerged and FALSE if it is floating.

NNDFALT is the default value of NN, the number of coefficients calculated. For the Ith cross section, if $|NNA(I)| >$ NNDFALT (see NNA(I) description, Line c) the first NN = NNDFALT coefficients are found in a single iteration process; then NN is increased by 1 per iteration process, using the previous set as initial values, until NN = $|NNA(I)|$ parameters have been determined. If left blank, NNDFALT=8 is assumed.

PLOT = TRUE results in the plotting subroutine SHIPLOT being called (see Line d). Blank is interpreted as FALSE.

READ STATEMENTS:*

a. READ 5, (TITLE(J), J = 1, 8)

b. READ 6, SUBMERG, NNDFALT, PLOT, AIPRINT, SPVOL **

c. READ 15, (NS(I), FRAME(I), SCALE(I), NNA(I), NDA(I), I = 1, III)

Note: III is the total number of ship cross sections appearing on DATA1 ($III \leq 24$)

d. If PLOT is TRUE: READ 20, (IPLOT(I), I = 1, 10), MULTI, FMAG, XEND, YEND

e. If the ship is floating, i.e., SUBMERG is FALSE:

  e1. READ 35, LBP, NSEGS, EQCAL

  e2. If EQCAL is FALSE: READ 40, WATERLN

  e3. If EQCAL is TRUE: READ 110, COLWID, WSCALE, ZSCALE, FLOODED

    e3i. If FLOODED is TRUE: READ 135, NXTRA, (ZXTRA(J), WXTRA(J), J = 1, NXTRA)

f. READ 75, NCASES, (NSPICK(J), J = 1, NCASES)

g. If NNA(I)<0 for the Ith section being processed: READ 105, IN, (AI(J), J = 1, IN)

Note: This read statement is executed once for each section having a negative NNA(•) value read in through statement c above.

FORMAT STATEMENTS:*

5 FORMAT (8A10)	40 FORMAT (E10.0)
6 FORMAT (L10, I10, 2L10, E10.0)	75 FORMAT (16I5)
15 FORMAT (I10, 2E10.0, 2I10)	105 FORMAT (I10/(8E10.0)
20 FORMAT (10I1, L10, 3E10.0)	110 FORMAT (3E10.0, L10)
35 FORMAT (E10.0, I10, L10)	135 FORMAT (I10/(2E10.0))

*These statements indicate the logical structure of INPUT file data and may differ from statements actually appearing in the program.

**Underlined variables have default values that are used if the data field is left blank (see definitions).

FIGURE 5. INPUT FILE READ STATEMENT STRUCTURE AND FORMATS

VARIABLE	DEFAULT VALUE	CONSEQUENCE
SUBMERG	FALSE	The ship is assumed to be on the surface.
NNDFALT	8	Eight mapping function coefficients are calculated.
PLOT	FALSE	SHIPLOT (plotting subroutine) is not called.
AIPRINT	FALSE	Computed coefficient values are not written on AIFILE.
SPVOL	35 FT^3/TON	These units are assumed. Saltwater is assumed.
SCALE(I)	1.	DATA1 section dimensions are not rescaled.
NNA(I)	0	NNDFALT parameters are calculated.
NDA(I)	0	*Fast-fitting* scheme for coefficients is used.
MULTI	FALSE	All chosen sections appear on a single plot.
FMAG	1.0	Y axis of plot is six inches long.
XEND	30.	X-axis length from ship origin is 30 feet.
YEND	30.	Y-axis length from ship orgin is 30 feet.
WSCALE	1.	Weight/length data on DATA2 are not rescaled.
ZSCALE	1.	Distance data on DATA2 are not rescaled.
FLOODED	FALSE	Ship is watertight. No additional weights read.
NCASES	0	Added masses are calculated for all sections.

FIGURE 6. DEFAULT VALUES FOR SHIPFIT INPUT VARIABLES

AIPRINT = TRUE results in output printed on AIFILE consisting of fitted parameter values (see Line f below). Blank is interpreted as FALSE.

SPVOL is the specific volume of seawater, usually expressed in ft^3/ton. If left blank (i.e., 0), a default value of 35 ft^3/ton, that is appropriate for standard seawater, is used. (Fresh water value is 36 ft^3/ton.)

Line c (Control Parameters for Specific Sections):

NS(I) is the station number of the section. Station 0 is usually located at the forward perpendicular (see Line f below).

FRAME(I) is the frame number of the section. Frame numbers are not used in calculations, but appear in the printout and plots.

SCALE(I) is a length scale factor that multiplies both DATA1 input coordinates. Blank implies a default value of one.

NNA(I) is the number of mapping function coefficients to be found for the section. The program storage limits allow for a maximum of 24 parameters. If NNA(I) = 0, NNDFALT is used. If NNA(I) < 0, initial values for the coefficients are read in by the user (see Line g below).

NDA(I) = 0, if the fast approximation of the Newton-Raphson iteration process described above is to be used initially. If convergence does not occur in 50 iterations, the exact method follows automatically. Any other value will result in the exclusive use of the exact method.

Line d (Plotting Control Parameters):

IPLOT(I) is assigned an integer value of 1 through 9 that controls the darkness of the plotted line or symbol for the Ith plot category described below. A zero value causes the Ith plot category to be

skipped. Lower values produce lighter lines, and higher values produce darker lines. A line of standard thickness is produced by the value 4. Available options are:

IPLOT(1) ≠ 0 ... Draw fitted spline function curve,
IPLOT(2) ≠ 0 ... Plot input points defining section (plotting symbol is a circle),
IPLOT(3) ≠ 0 ... Draw fitted mapping function curve,
IPLOT(4) ≠ 0 ... Plot Z-plane images of points on unit circle (plotting symbol is a triangle).

(Variables IPLOT(5) through IPLOT (10) are reserved for future use.)

MULTI = TRUE results in multiple plots -- one for each section picked in Line f. FALSE results in all picked sections being plotted on a single plot (see example in Figure 13, p. 39).

FMAG is a magnification factor that can be used to shrink or increase the size of the plots, e.g., FMAG = .5 produces a half size plot, while the plot resulting from FMAG = 1.5 would be 50 percent larger than the standard size which has a y-axis length of six inches. The default value is 1.0.

XEND is the distance in feet from the ship origin to the end of the plotted x axis. It must be a multiple of 10. The default value is 30.

YEND is the distance in feet from the skip origin to the end of the plotted y axis. It must be a multiple of 10. The default value is 30.

Line e1 (Information Required If the Ship Is Floating on the Surface):

LBP is the ship length between perpendiculars (length at the designer's waterline) in feet. LBP is REAL.

NSEGS is the total number of ship segments (used to calculate segment length, LBP/NSEGS). Default value is 20.

EQCAL = TRUE, if the equilibrium position is to be calculated. Blank is interpreted as FALSE.

Line e2 (Waterline to be Used for Surfaced Ship):

WATERLN is the waterline position relative to the data origin in feet.

Line e3 (Parameters for Determining Weight Histogram from DATA2 Input):

COLWID is the column width of the weight histogram in feet.

WSCALE is a scale factor that multiplies the weight per unit length data read in from DATA2. Blank implies a default value of one.

ZSCALE is a scale factor that multiplies the weight histogram column position data. Blank implies a default value of one.

FLOODED is TRUE if the ship has been flooded and extra lumped weights are to be read from the INPUT file.

Line e3i (Additional Weight Distribution):

NXTRA is the total number of extra lumped weights and weight positions to be read. NXTRA $\leq$ 24.

ZXTRA is an array of extra lumped weight positions in feet relative to the forward perpendicular with the same sign convention as in the weight histogram (zero at the forward perpendicular and positive in the direction of the after perpendicular).

WXTRA is an array of lumped weights in (long) tons.

Line f (Sections Chosen For Added Mass Computations):

NCASES is the number of cross sections of DATA1 to be processed. If NCASES = 0 (Blank), all forms are processed in the order of their appearance on DATA1.

NSPICK is an array of section station numbers in the order in which processing is to occur.

Line g (Optional Input of Initial Coefficient Values):

IN is the number of initial coefficient values to be read in. (If the number NN of coefficients to be calculated is greater than IN, the additional NN-IN coefficients are given initial values of zero.)

AI is an array of initial coefficient values. Note that if NNA(I) $\geq$ 0 for the Ith section, initial coefficient values are generated internally by the program.

INPUT File Example

Figure 7 shows the INPUT file records for a sample problem concerning a class of destroyers. (The examples shown in Figures 3 and 4 also pertain to this ship design.) In Figure 7, considerable use is made of the default features built into SHIPFIT. The blank fields on Card 2 indicate that the ship is floating (SUBMERG is FALSE), the default value for the number of mapping function coefficients is 8 (NNDFALT = 8), no printing of final coefficient values on AIFILE is desired (AIPRINT if FALSE), and seawater is assumed (SPVOL = 35 ft^3/ton). Cards 3 through 17 give station and frame numbers. Here SCALE(I), NNA(I), and NDA(I) take their default values indicated in Figure 6. Card 18 is needed because PLOT is TRUE in Card 2. Card 18 indicates that spline functions will be drawn with a standard line thickness (IPLOT(1) = 4 in column 1) and input points will be plotted with somewhat heavier symbols (IPLOT(2) = 6 in column 2). Individual plots will be prepared

SHIPFIT SAMPLE PROBLEM						CARD 1
		TRUE				CARD 2
3	33.					CARD 3
4	44.					CARD 4
5	55.					CARD 5
6	66.					CARD 6
7	77.					CARD 7
8	88.					CARD 8
9	99.					CARD 9
10	109.					CARD 10
11	112.					CARD 11
12	123.					CARD 12
13	134.					CARD 13
14	145.					CARD 14
15	156.					CARD 15
16	167.					CARD 16
17	178.					CARD 17
46	TRUE	.75	30.	20.		CARD 18
383.						CARD 19
19.						CARD 20
1 3						CARD 21

12345678901234567890123456789012345678901234567890123456789012345678901234567890

FIGURE 7. INPUT FILE RECORDS FOR SURFACE SHIP SAMPLE PROBLEM

for each section selected (MULTI is TRUE), and a 75 percent reduced plot will be produced with x and y axis lengths of 30 and 20 feet, respectively. Card 19 shows the length between perpendiculars LBP, with 20 segments assumed by default. Because EQCAL is blank (FALSE) on Card 19, Card 20 must supply the position of the waterline relative to the data origin used for the DATA1 file. Finally, Card 21 says one section is to be analyzed (NCASES=1) and the selected section is at station number 3 (NSPICK(1) = 3).

The output produced by this sample problem is shown in Figures 8, 9, and 11 and is discussed below.

Output

SHIPFIT produces three different kinds of output: printout, plots (optional), and a file of calculated mapping function coefficient values called AIFILE (optional). The principal form of output is the printout. A page of information detailing the various aspects of the added mass calculation is printed out for each selected section. An example produced by the input of Figure 3, 4, and 7 is shown in Figure 8. Results included are the residuals of the fit of the spline function to the input points, the RMS error of the spline fit, initial and final values of the mapping function coeffients showing individual and overall quadratic mean errors for the fitted coefficients, the calculated ship form as represented by the Z-plane images of points equally spaced about the unit circle, differences of colinear radii drawn from the calculation origin to the image points and to the spline function curve, the submerged beam and area, the buoyancy coefficient (this is the ratio of the submerged area to the area within a semicircle with a diameter equal to the submerged beam), the draft, and horizontal and vertical added masses per unit length and added mass coefficients. The distinction between the "best" and "check" values of the vertical added mass is explained in the following section on program operational guidelines.

If the ship is floating on the surface, the pages describing the added mass calculations are preceded by a page of ship orientation information. Formats of this page differ, depending upon whether the waterline is read in from the INPUT

```
SHIPFIT SAMPLE PROBLEM

CALCULATION OF ADDED MASSES

STATION NUMBER    3          LOCATED AT FRAME  33.00

     RESIDUALS FOR SPLINE FIT TO 23 INPUT POINTS WITH 3 POINTS PER ARC OF SPLINE
         .835E-02   -.283E-01    .458E-01   -.355E-01    .341E-02    .877E-02    .976E-02   -.131E-01   -.200E-01    .327E-01
         .229E-02   -.360E-01    .198E-01    .311E-02    .139E-01   -.281E-01    .937E-02    .896E-02    .521E-02   -.349E-01
         .409E-01   -.244E-01    .807E-02

     RMS ERROR OF SPLINE FIT =    .3725E-01

     INITIAL MAPPING PARAMETERS
      14.9681197              4.0242337              0.0000000              0.0000000              0.0000000
       0.0000000              0.0000000              0.0000000

     FINAL MAPPING PARAMETERS AFTER  5. ITERATIONS   ( QME )          MODE 0
      14.3300988( .006448)    -4.2389115( .009227)     .3219726( .008865)     .0683421( .008685)     .1953563( .008346)
        .0264536( .008407)      .0886466( .008436)     .0602093( .008437)

     QME OF OVERALL FIT  .0312340

     CALCULATED SHIP FORM
         X       Y      RM-RS         X       Y      RM-RS         X       Y      RM-RS         X       Y      RM-RS
       -.00  -19.02    .0302         .58  -18.95   -.0385        1.18  -18.74   -.0069        1.80  -18.39    .0089
       2.47  -17.92    .0100        3.19  -17.38    .0056        3.94  -16.80   -.0106        4.67  -16.20   -.0132
       5.33  -15.57    .0015        5.91  -14.85    .0162        6.42  -14.01    .0067        6.91  -13.05   -.0095
       7.41  -12.00   -.0165        7.91  -10.92   -.0059        8.37   -9.82    .0144        8.76   -8.68    .0216
       9.07    7.47    .0009        9.34   -6.15   -.0273        9.66   -4.76   -.0338       10.04   -3.39    .0039
      10.43   -2.13    .0632       10.74   -1.01    .0652       10.85    -.00   -.0893

     SUBMERGED BEAM =  21.89

     SUBMERGED AREA = 293.09

     BUOYANCY COEFFICIENT = 1.55789

     SECTION DRAFT =  18.99

                                     VERTICAL       HORIZONTAL

     ADDED MASS /L  (BEST)             4.61            6.69

     ADDED MASS /L  (CHECK)            4.60

     ADDED MASS COEFFICIENT           .85689          .41329
```

FIGURE 8. PRINTOUT OF ADDED MASS INFORMATION FOR SAMPLE PROBLEM

file or calculated by balancing the forces and torques that bear on the ship. Figures 9 and 10 show these two forms of the output. Figure 9 is the result of the input given in Figure 7. An additional column of extra weights is included in Figure 10 if these are present in the INPUT file.

Forms of the plotted output are shown in Figures 11, 12, and 13. Figures 11 and 12 show the individual plots that are produced when the variable MULTI is TRUE, while Figure 13 is the type of result obtained when MULTI is FALSE. Figure 11 was produced by the INPUT file shown in Figure 7. It is a plot of the input points and fitted spline function for the cross section located at station 3. Figure 12 shows, in addition to the spline function, plots of the image points (triangles) and the fitted mapping function.

Program Operation Guidelines

It is apparent from the description of the INPUT file that SHIPFIT allows some flexibility in the manner in which the parameter fitting is carried out. Although the program-assigned set of initial coefficient values given in Equations (42) and (43) usually are sufficient to produce convergence, an occasional section form may be encountered where these conditions fail to produce convergence. For such forms it may then be necessary to set NNDFALT to a value smaller than the desired number of coefficients NNA (II) where convergence is guaranteed (such as 6 or 8) and obtain the desired NNA (II) coefficients incrementally. A single form that is difficult to fit can be worked on independently by using the NSPICK array. The incremental fitting of parameters can be done either automatically as described in the NNDFALT definition (Line b), or in a single-step-at-a-time procedure using the AI array, and where NNDFALT is incremented in successive runs.

A problem in fitting may occur along concave portions of a particular ship contour as shown in Figure 12. Points that are equally spaced on the unit circle tend to map into points that are more tightly spaced along convex lengths of the ship contour and less closely spaced along concave parts. This presents a fitting problem that could be remedied by locating more points in regions of the unit circle that map into the concave regions of the contour. Presently,

```
   1 *0*9X*ADDED MASS /L  (BEST)*7X,F12.2,5X,F12.2)
     PRINT 294,AV2
 294 FORMAT(*0*9X*ADDED MASS /L  (CHECK)*6X,F12.2)
     PRINT 295,CV,CH
 295 FORMAT(*0*9X*ADDED MASS COEFFICIENT*6X,F12.5,5X,F12.5)
     IF(PLOT) CALL SHIPLOT(2)
 300 CONTINUE
 400 IF(PLOT) CALL SHIPLOT(3)
     STOP
     END
```

```
  264 FORMAT(10X,2F7.2,F8.4,3X,2F7.2,F8.4,3X,2F7.2,F8.4,3X,2F7.2,F8.4)
C
C*** CALCULATE CROSS SECTIONAL AREAS FROM MAPPING FUNCTION COEFFICIENTS
      SUM=0.
  270 DO 275 K=1,NN
      ABK=ALPHA-BETA*FLOAT(K)
      SUM=SUM+ABK*AA(K,1)**2
  275 CONTINUE
      AREAMF=PI*SUM
      IF(.NOT.SUBMERG) AREAMF=AREAMF/2.
C
C*** CALCULATE HYDRODYNAMIC COEFFICIENTS FOR SECTION
      BEAM=BD2*2.
      CBUOY=2.*AREA(II)/PI/BD2**2
      IF(SUBMERG) CBUOY=CBUOY/2.
      PRINT 280,BEAM,AREA(II),CBUOY
  280 FORMAT(*0*9X*SUBMERGED BEAM =*F7.2//10X*SUBMERGED AREA =*F7.2//
     1 10X*BUOYANCY COEFFICIENT =* F10.5)
      IF(SUBMERG) GO TO 290
      AV1=DENSITY*(PI*AA(1,1)*(AA(1,1)+AA(2,1))-AREA(II))
      AV2=DENSITY*(PI*AA(1,1)*(AA(1,1)+AA(2,1))-AREAMF)
      AVCYL=DENSITY*PI*BD2**2/2.
      CV=AV1/AVCYL
      SUM1=SUM2=SUM3=0.
      N1=NN-1
      DO 286 IV=2,N1
      SUM1=SUM1+AA(IV+1,1)*CC(1,IV)
      SUM2=SUM2+AA(IV+1)**2*CC(IV,IV)
      IF(IV.EQ.2) GO TO 286
      IV1=IV-1
      DO 284 IU=2,IV1
      SUM3=SUM3+AA(IU+1,1)*AA(IV+1,1)*CC(IU,IV)
  284 CONTINUE
  286 CONTINUE
      AH=8.*DENSITY/PI*((AA(2,1)-AA(1,1))**2*CC(1,1)
     1+2.*(AA(2,1)-AA(1,1))*SUM1+SUM2+2.*SUM3)
      DRAFT=YCSO(II)-YCL(II)
      AHCYL=DENSITY*PI*DRAFT**2/2.
      CH=AH/AHCYL
      PRINT 288,DRAFT
  288 FORMAT(*0*9X*SECTION DRAFT =*F7.2)
      GO TO 292
  290 AV1=DENSITY*(2.*PI*AA(1,1)*(AA(1,1)+AA(3,1))-AREA(II))
      AV2=DENSITY*(2.*PI*AA(1,1)*(AA(1,1)+AA(3,1))-AREAMF)
      AVCYL=DENSITY*PI*BD2**2
      CV=AV1/AVCYL
      AH=DENSITY*(2.*PI*AA(1,1)*(AA(1,1)-AA(3,1))-AREA(II))
      H=DCU(II)-YCL(II)
      HD2=H/2.
      AHCYL=DENSITY*PI*HD2**2
      CH=AH/AHCYL
      PRINT 291,H
  291 FORMAT(*0*9X*SECTION HEIGHT =*F7.2)
  292 PRINT 293,AV1,AH
  293 FORMAT(*0*T44*VERTICAL*T59*HORIZONTAL*/
```

```
      NDATA(5)=0
C
C*** CALCULATE CX AND CY FUNCTIONS OF ANGLES IN ZETA PLANE
      DO 220 M=1,MM
      THETAM=THSTP*FLOAT(M-1)-PID2
      DO 220 N=1,NN
      ARG=(ALPHA-BETA*FLOAT(N))*THETAM
      CARG=COS(ARG)
      SARG=SIN(ARG)
      IF(SUBMERG) GO TO 214
      CX(N,M)=CARG
      CY(N,M)=SARG
      GO TO 220
  214 IF(MOD(N,2).EQ.0) GO TO 216
      CX(N,M)=CARG
      CY(N,M)=SARG
      GO TO 220
  216 CX(N,M)=-SARG
      CY(N,M)=CARG
  220 CONTINUE
C
C*** CALCULATE MAPPING FUNCTION PARAMETERS USING LS CRITERION
      N=NN
      IF(NN.LE.NNDFALT) GO TO 224
      N=NNDFALT
      IF(IN.NE.0) N=IN
  224 NDATA(6)=NDA(II)
  230 NDATA(1)=N
  240 DO 242 K=1,NN
      AA(K,1)=AI(K)
  242 CONTINUE
      CALL LSQSUB(NDATA,X,AA,LSQFUN,1.E-5)
      IF(AA(26,1).NE.0.) GO TO 244
      IF(NDATA(6).EQ.1) GO TO 250
      NDATA(6)=1
      GO TO 240
  244 PRINT 245,AA(26,1),NDATA(6),(AA(I,1),AA(I,2),I=1,N)
  245 FORMAT(1H0, 9X*FINAL MAPPING PARAMETERS AFTER *F3.0* ITERATIONS
     +( QME )*10X*MODE*I2/(7X,5(3X,F11.7*(*F8.6*)*)))
      IF(AIPRINT) PRINT(3,105) N,(AA(I,1),I=1,N)
      N=N+1
      IF(N.LE.NN) 230,256
  250 PRINT 255,NDATA(4),(AA(I,1),AA(I,2),I=1,NN)
  255 FORMAT(1H0,1X*NO CONVERGENCE. MAPPING PARAMETERS AFTER*I3* ITERATI
     +ONS   ( QME )*/(7X,5(3X,F11.7*(*F8.6*)*)))
  256 PRINT 257,AA(26,2)
  257 FORMAT(1H0,9X*QME OF OVERALL FIT*F10.7)
C
C*** PRINT CALCULATED SHIP FORM IN COORDINATES RELATIVE TO SHIP ORIGIN (SO)
      PRINT 260
  260 FORMAT(*0*9X*CALCULATED SHIP FORM*/14X,4(*X*6X*Y*5X*RM-RS*7X))
      DO 262 M=1,MM
      X(8,M)=X(9,M)-X(2,M)
  262 CONTINUE
      PRINT 264,(X(6,M),X(7,M),X(8,M),M=1,MM)
```

```
  110 NN=NNA(II)
      GO TO 120
  115 NN=NNDFALT
  120 DO 130 J=1,JJ
      P=PCIN(J,II)
      CALL SPLCAL
      DIF(J)=RCIN(J,II)-R
  130 CONTINUE
      PRINT 135,JJ,KK,(DIF(J),J=1,JJ)
  135 FORMAT(1H0,9X*RESIDUALS FOR SPLINE FIT TO*I3* INPUT POINTS WITH*I2
     + * POINTS PER ARC OF SPLINE*/(10X,10G12.3))
      PRINT 140,E(II)
  140 FORMAT(1H0,9X*RMS ERROR OF SPLINE FIT =*G12.4)
C
C*** PREPARE COEFFICIENTS AND CONSTANTS FOR FITTING MAPPING FUNCTION
      MM=2.5*FLOAT(NN)
      IF(MM.LT.JJ) MM=JJ
      THSTP=PI/FLOAT(MM-1)
      IF(SUBMERG) GO TO 160
      THSTP=THSTP/2.
  150 ALPHA=3.
      BETA=2.
      IF(IN.NE.0) GO TO 170
      XLIM=DCU(II)
      BD2=XLIM-XCSO(II)
      D=YCSO(II)-YCL(II)
      AI(1)=(D+BD2)/2.
      AI(2)=(D-BD2)/2.
      JN=3
      GO TO 170
  160 ALPHA=2.
      BETA=1.
      CALL SUBBEAM(BD2)
      IF(IN.NE.0) GO TO 170
      YSEMI=(DCU(II)-YCL(II))/2.
      AI(1)=(YSEMI+AREA(II)/PI/YSEMI)/2.
      AI(2)=(DCU(II)+YCL(II))/2.-YCSO(II)
      AI(3)=AI(1)-YSEMI
      JN=4
  170 IF(IN.NE.0) JN=IN+1
      IF(JN.GT.NN) GO TO 185
      DO 180 N = JN,NN
      AI(N)=0.
  180 CONTINUE
  185 DO 190 N=1,NN
      AA(N,3)=N
  190 CONTINUE
      PRINT 200,(AI(N),N=1,NN)
  200 FORMAT(1H0,9X*INITIAL MAPPING PARAMETERS*/(7X,5(3X,F11.7,10X)))
      DO 210 M=1,MM
      X(1,M)=M
  210 CONTINUE
      NDATA(2)=MM
      NDATA(3)=2
      NDATA(4)=50
```

```
      JJ=JJA(I)
      E(I)=0.
      XCSO(I)=SGN(I)*XCSO(I)
      KKA(I)=(JJA(I)+22)/23+1
      IF(KKA(I).LT.3) KKA(I)=3
      DO 40 J=1,JJ
      IF(SGN(I).EQ.-1) PCIN(J,I)=SIGN(PI,PCIN(J,I))-PCIN(J,I)
      IF(J.EQ.1) GO TO 40
      IF(PCIN(J,I).GT.PCIN(J-1,I)) GO TO 40
      PRINT 35, NS(I),J
   35 FORMAT(*0+++++++ IMPROPER DATA FORMAT OR CALCULATION ORIGIN FOR SE
     +CTION*I5*    NEAR POINT NUMBER*I5* ... STOP ++++++++++*)
      STOPPER =1.
   40 CONTINUE
   50 CONTINUE
      IF(STOPPER.EQ.1.) STOP "BY SHIPFIT"
      IF(PLOT) CALL SHIPLOT(1)
C
C*** FIT CUBIC SPLINE FUNCTIONS TO CO REFERENCED DATA
      DO 70 I=1,III
      JJ=JJA(I)
      CALL SPLSQ1(JJ,PCIN(1,I),RCIN(1,I),CF(1,I),E(I),KKA(I))
      DO 60 J=1,JJ
      RC2(J,I)=RCIN(J,I)**2
   60 CONTINUE
      CALL SPLSQ1(JJ,PCIN(1,I),RC2(1,I),CA(1,I),-1.,KKA(I))
   70 CONTINUE
C
C*** CALCULATE SUBMERGED CROSS-SECTIONAL AREAS AND SET LOCATION OF SHIP ORIGIN.
C    IF VESSEL IS FLOATING FIND EQUILIBRIUM POSITION. OBTAIN LIMITS.
      CALL ORIENT
C
C*** CALCULATE ADDED MASSES OF SECTIONS
      READ 75,NCASES,(NSPICK(J),J=1,NCASES)
   75 FORMAT(16I5)
      DO 300 IS=1,III
      II=IS
      IF(NCASES.EQ.0) GO TO 90
      DO 80 J=1,NCASES
      IF(NS(II).EQ.NSPICK(J)) GO TO 90
   80 CONTINUE
      GO TO 300
   90 JJ=JJA(II)
      KK=KKA(II)
      XSHIFT=XCSO(II)
      YSHIFT=YCSO(II)
      PRINT 95,(TITLE(J),J=1,8),NS(II),FRAME(II)
   95 FORMAT(*1*4X8A10//5X*CALCULATION OF ADDED MASSES*//
     1 5X*STATION NUMBER*I5,10X*LOCATED AT FRAME*F7.2)
      IN=0
      IF(NNA(II))100,115,110
  100 NN=-NNA(II)
      READ 105,IN,(AI(I),I=1,IN)
  105 FORMAT(I10/(8G10.4))
      GO TO 120
```

```
      PROGRAM SHIPFIT(INPUT,OUTPUT,DATA1,DATA2,AIFILE,
     1 TAPE1=DATA1,TAPE2=DATA2,TAPE3=AIFILE)
      DIMENSION NDATA(6),NNA(24),SGN(24),DIF(64),NSPICK(24),
     1 AI(24),NDA(24)
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      COMMON /BLOCK2/ AA(26,3),TITLE(8),XSHIFT,YSHIFT,ALPHA,BETA,
     1 CX(24,64),CY(24,64),MM,NN,NSEGS
      COMMON /BLOCK3/ PID180,YDCO(24),CA(28,24),RC2(64,24),X(9,64)
      EXTERNAL LSQFUN
      LOGICAL SUBMERG,AIPRINT,PLOT
      DATA PI,PID2/3.1415926535898,1.5707963267949/
      PID180=PI/180.
      READ 5,(TITLE(I),I=1,8)
    5 FORMAT(8A10)
      READ 6,SUBMERG,NNDFALT,PLOT,AIPRINT,SPVOL
    6 FORMAT(L10,I10,2L10,E10.0)
C
C*    SET DEFAULT VALUE OF NNDFALT.
      IF(NNDFALT.EQ.0) NNDFALT=8
C*    SET DEFAULT VALUE OF SPVOL TO 35 CU FT / TON (LONG)
      IF(SPVOL.EQ.0.) SPVOL=35.
      DENSITY=1./SPVOL
C
C*** READ AND CONVERT X,Y SHIP FORM DATA FROM DATA ORIGIN (DO)
C                    TO CALCULATION ORIGIN (CO)
      III=0
   10 READ(1,*) IDIG,XDIG,YDIG
      IF(EOF(1).NE.0) GO TO 30
      IF(IDIG.NE.1) GO TO 20
      III=III+1
      READ 15,NS(III),FRAME(III),SCALE,NNA(III),NDA(III)
   15 FORMAT(I10,2E10.0,2I10)
C*    SET DEFAULT VALUE OF SCALE TO 1.
      IF(SCALE.EQ.0.) SCALE=1.
      JJA(III)=0
      XDMIN=1.E10
      XDMAX=-1.E10
      XDCO=XDIG*SCALE
      YDCO(III)=YDIG*SCALE
      XCSO(III)=-XDCO
      GO TO 10
   20 JJ=JJA(III)=JJA(III)+1
      XD=XDIG*SCALE
      XDMIN=AMIN1(XD,XDMIN)
      XDMAX=AMAX1(XD,XDMAX)
      SGN(III)=SIGN(1.,XDMIN+XDMAX)
      XC=XD-XDCO
      YC=YDIG*SCALE-YDCO(III)
      RCIN(JJ,III)=SQRT(XC**2+YC**2)
      PCIN(JJ,III)=ATAN2(YC,XC)
      GO TO 10
   30 STOPPER=0.
      DO 50 I=1,III
```

APPENDIX A

LISTING OF THE <u>SHIPFIT</u> COMPUTER PROGRAM
(SUBPROGRAMS IN ALPHABETICAL ORDER)

REFERENCES

1. Landweber, L., and Macagno, M. C., "Added Mass of Two-Dimensional Forms Oscillating in a Free Surface," Journal of Ship Research, Vol. 1, Nov 1957, pp. 36-48.

2. von Kerczek, C., and Tuck, E. O., "The Representation of Ship Hulls by Conformal Mapping Functions," Journal of Ship Research, Dec 1969, pp. 284-298.

however, there is no special point allocation feature of this kind built into SHIPFIT. Satisfactory results can usually be achieved by simply increasing the number of mapping function coefficients desired, since the number of image points is increased proportionately.

For all difficult sections, plotting of the input points and mapping function is recommended. Another useful check of the level of success achieved in fitting the mapping function to the spline function is provided by a "check" value of the vertical added mass. This is printed automatically for all cross sections. The "check" value is obtained by using a cross-sectional area calculated from the full set of fitted mapping coefficients, while the value labeled "best" is obtained by using an area calculated from the cubic spline function. A discrepancy between the "best" and "check" values indicates a problem in fitting the mapping function coefficients. Most sections will be fitted easily, without the need for the special features built into SHIPFIT. Additional form data that may be used as test cases for the program can be found in Reference 1.

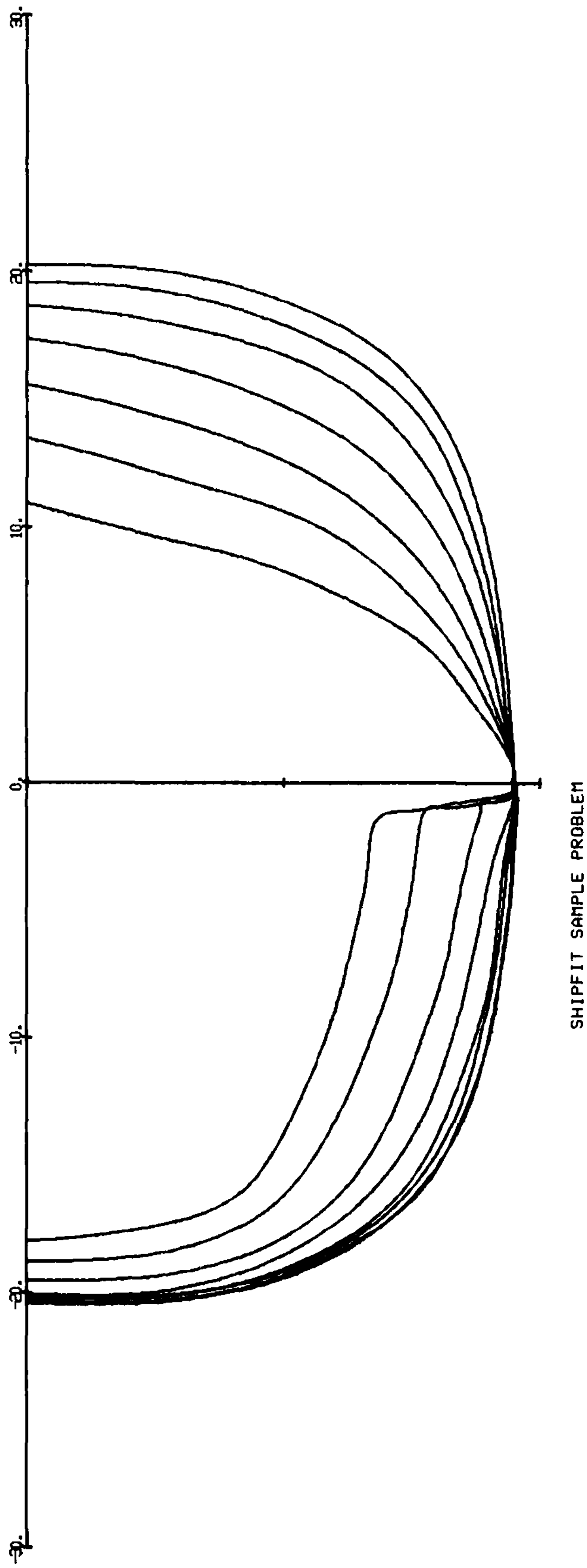


FIGURE 13. PLOT OF SUPERIMPOSED SECTIONS (MULTI VARIABLE IS FALSE)

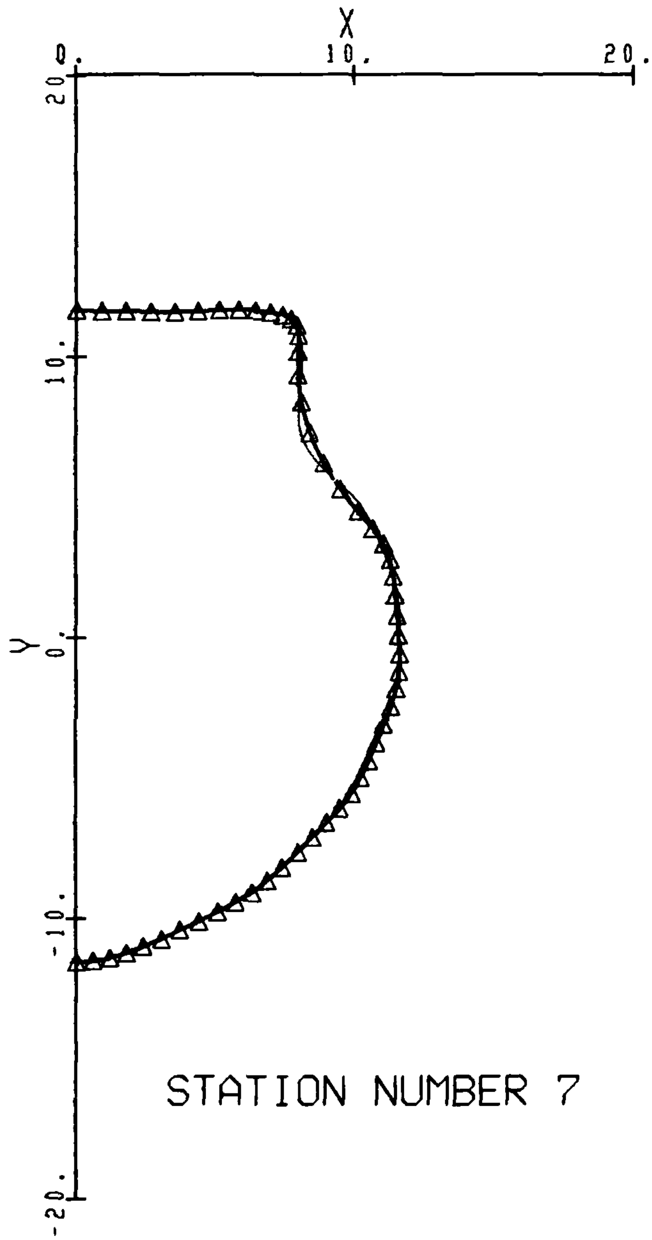


FIGURE 12. PLOT OF SPLINE FUNCTION, MAPPING FUNCTION, AND IMAGE POINTS FOR A SUBMERSIBLE SECTION

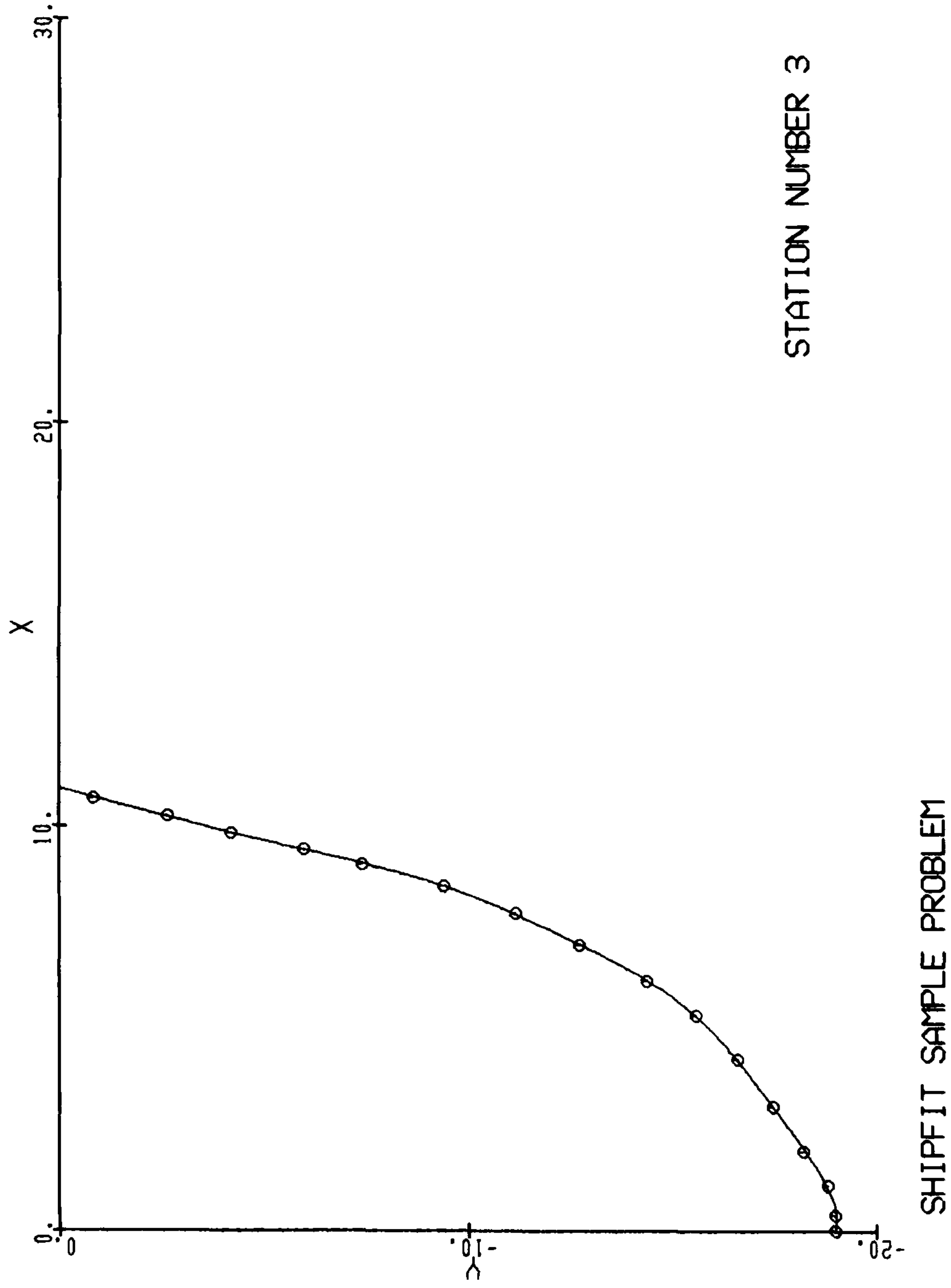


FIGURE 11. PLOT OF INPUT POINTS AND SPLINE FUNCTION PRODUCED BY INPUT DATA OF FIGURE 7

SHIPFIT SAMPLE PROBLEM

RESULTS OF SHIP EQUILIBRIUM CALCULATION ... (UNLISTED STATIONS CONSIDERED NEUTRAL)

EXTRA WEIGHT DUE TO FLOODING = 0.00

DISPLACEMENT = 3440.70

ANGLE OF KEEL = .32 DEGREES FROM HORIZONTAL (POS = BOW UP)

STATION	FRAME	FREEBOARD	DRAFT	DIST FROM STA 0	BUOYANCY	W(DATA2)
				9.70		33.32
				28.84		48.10
				47.52		99.29
3	33.0	15.26	13.29	57.45	96.82	
				67.11		90.91
4	44.0	14.95	13.40	76.60	125.05	
				86.14		150.92
5	55.0	13.87	13.57	95.75	152.14	
				105.10		219.84
6	66.0	12.75	13.56	114.90	182.85	
				124.60		274.23
7	77.0	11.83	13.71	134.05	212.72	
				143.60		205.10
8	88.0	10.86	13.83	153.20	233.28	
				162.40		194.56
9	99.0	10.08	13.93	172.35	251.59	
				181.50		221.57
10	109.0	9.37	14.05	191.50	261.80	
				200.60		261.59
11	112.0	9.61	14.14	210.65	265.20	
				219.60		193.61
12	123.0	9.12	14.20	229.80	259.68	
				238.40		209.50
13	134.0	8.88	14.36	248.95	257.81	
				257.20		212.18
14	145.0	8.25	14.40	268.10	241.82	
				276.50		279.97
15	156.0	7.61	14.62	287.25	218.63	
				295.80		246.08
16	167.0	7.94	14.59	306.40	178.21	
				315.10		193.61
17	178.0	7.84	14.72	325.55	143.89	
				334.40		148.76
				354.20		80.77
				373.40		76.79

FIGURE 10. CALCULATED SHIP ORIENTATION FOR SAMPLE PROBLEM DATA

SHIPFIT SAMPLE PROBLEM

SHIP EQUILIBRIUM POSITION NOT CALCULATED ... PROGRAM USES INPUT WATERLINE

INPUT WATERLINE (RELATIVE TO DATA ORIGIN) = 19.00

TOTAL BUOYANCY FORCE = 4532.29

STATION	FRAME	FREEBOARD	DRAFT	DIST FROM STA 0	BUOYANCY
3	33.0	9.56	18.99	57.45	160.36
4	44.0	9.35	19.00	76.60	203.22
5	55.0	8.38	19.05	95.75	241.92
6	66.0	7.37	18.95	114.90	282.50
7	77.0	6.55	18.99	134.05	318.95
8	88.0	5.69	19.00	153.20	343.42
9	99.0	5.01	18.99	172.35	363.52
10	109.0	4.41	19.00	191.50	372.31
11	112.0	4.76	19.00	210.65	373.64
12	123.0	4.37	18.95	229.80	364.64
13	134.0	4.24	19.00	248.95	360.38
14	145.0	3.71	18.94	268.10	341.46
15	156.0	3.18	19.05	287.25	313.16
16	167.0	3.62	18.92	306.40	266.79
17	178.0	3.62	18.94	325.55	226.01

FIGURE 9. SHIP ORIENTATION INFORMATION PAGE FOR SAMPLE PROBLEM

```
      SUBROUTINE AREACAL
C*** AREACAL CALCULATES THE SUBMERGED CROSS-SECTIONAL AREA USING THE POLAR
C    COORDINATE SYSTEM CENTERED AT THE CALCULATION ORIGIN (CO).
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      COMMON /BLOCK3/ PID180,YDCO(24),CA(28,24),RC2(64,24),X(9,64)
      LOGICAL SUBMERG
      DATA PI,PID2/3.1415926535898,1.5707963267949/
      IF(SUBMERG) 10,30
   10 AY=0.
      IF(XCSO(II).NE.0.) GO TO 20
      P=PCL=-PID2
      CALL SPLCAL
      YCL(II)=-R
      P=PCU=PID2
      CALL SPLCAL
      DCU(II)=R
      AX=0.
      GO TO 80
   20 CALL YLIMIT(PCIN(1,II),YCL(II))
      CALL YLIMIT(PCIN(JJ,II),DCU(II))
      PCL=ATAN2(YCL(II),XCSO(II))
      PCU=ATAN2(DCU(II),XCSO(II))
      AX=XCSO(II)*(DCU(II)-YCL(II))/2.
      GO TO 80
   30 IF(XCSO(II).NE.0.) GO TO 40
      P=PCL=-PID2
      CALL SPLCAL
      YCL(II)=-R
      AX=0.
      GO TO 50
   40 CALL YLIMIT(PCIN(1,II),YCL(II))
      PCL=ATAN2(YCL(II),XCSO(II))
      AX=XCSO(II)*(YCSO(II)-YCL(II))/2.
   50 IF(YCSO(II).NE.0.) GO TO 60
      P=PCU=0.
      CALL SPLCAL
      DCU(II)=R
      AY=0.
      GO TO 80
   60 DO 65 J=1,JJ
      IF(RCIN(J,II)*SIN(PCIN(J,II))-YCSO(II).GT.0.) GO TO 70
   65 CONTINUE
      J=JJ
   70 CALL XLIMIT(PCIN(J,II),DCU(II))
      PCU=ATAN2(YCSO(II),DCU(II))
      AY=YCSO(II)*(XCSO(II)-DCU(II))/2.
   80 CALL SPLINT(JJ,PCIN(1,II),RC2(1,II),KKA(II),ACO2,PCL,PCU,CA(1,II))
      AREA(II)=2.*(ACO2/2.-AX-AY)
      RETURN
      END
```

```
      FUNCTION CC(IU,IV)
      IR=2*IU-1
      IS=2*IV-1
      IF(IR.EQ.IS) GO TO 30
      IF(IR.LT.IS) GO TO 10
      IX=IS
      IS=IR
      IR=IX
   10 FNF=NF=IS-IR
      SUM=0.
      DO 20 N=2,NF,2
      SUM=SUM+1./FLOAT(IR+N)/FLOAT(IS-N)
   20 CONTINUE
      CC=-IR*IS/2./FNF*SUM
      GO TO 50
   30 NF=2*IS
      SUM=0.
      DO 40 N=2,NF,2
      SUM=SUM+1./(FLOAT(IS-N))**2
   40 CONTINUE
      CC=IS/4.*SUM
   50 RETURN
      END
```

```
      SUBROUTINE LSQFUN(NDATA,X,D1,A,D2)
C*** LSQFUN CALCULATES THE MAPPING FUNCTION AND SPLINE FUNCTION RADII (RELATIVE
C    TO THE CALCULATION ORIGIN) AND THE DERIVATIVES OF THE RESIDUAL TERM AS
C    REQUIRED BY LSQSUB.
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      COMMON /BLOCK2/ AA(26,3),TITLE(8),XSHIFT,YSHIFT,ALPHA,BETA,
     1 CX(24,64),CY(24,64),NN
      DIMENSION NDATA(6),X(9),D1(1),A(1),DR(24),DP(24),D2(28,28)
      N1=NDATA(1)
      M=X(1)
      XM=YM=0.
      DO 10 N=1,N1
      XM=XM+A(N)*CX(N,M)
      YM=YM+A(N)*CY(N,M)
   10 CONTINUE
      X(6)=XM
      X(7)=YM
      XXC=XM+XSHIFT
      YYC=YM+YSHIFT
      RC2=XXC**2+YYC**2
      RC=SQRT(RC2)
      CX1=XXC/RC
      CY1=YYC/RC
      CX2=CX1/RC
      CY2=CY1/RC
      P=ATAN2(YYC,XXC)
      CALL SPLCAL
      X(2)=R
      X(9)=RC
      DO 20 N=1,N1
      DR(N)=CX1*CX(N,M)+CY1*CY(N,M)
      DP(N)=CX2*CY(N,M)-CY2*CX(N,M)
      D1(N)=DR(N)-RP*DP(N)
   20 CONTINUE
      IF(NDATA(6))25,40
   25 DO 30 L=1,N1
      DO 30 K=L,N1
      D2R=(CX(L,M)*CX(K,M)+CY(L,M)*CY(K,M)-DR(K)*DR(L))/RC
      D2P=(CX(L,M)*CY(K,M)-CY(L,M)*CX(K,M))/RC2-2.*DR(L)*DP(K)/RC
      D2(K,L)=D2R-RP*D2P-RPP*DP(K)*DP(L)
   30 CONTINUE
   40 RETURN
      END
```

```
      SUBROUTINE LSQSUB (N, X, A, VF, DELT)
C
C*** LSQSUB IS A GENERAL NON-LINEAR LEAST SQUARES FITTING  ROUTINE.
C     IT WAS WRITTEN BY TED ORLOW AND IS AVAILABLE THROUGH THE NSWC/WHITE
C     OAK PROGRAM LIBRARY.
C
      DIMENSION DF(25), A(26,3), X(9,1), V(28,28), N(5), D2F(28,28)
      NPAR = N(1)
      NPARM1=NPAR-1
      NPT  = N(2)
      NPCD = N(3)
      NWT = NPCD+1
      NWAR = N(5)
      NIT = 0
      IF(NPAR.LT.26.AND.NPAR.GT.0.AND.NPCD.LT.8.AND.NPCD.GT.1)GO TO 20
      PRINT 10, NPAR,NPCD
   10 FORMAT(1H0,6HNPAR =,I5,4X,2HOR,2X,6HNPCD =,I5,4X,
     115HIS OUT OF RANGE)
      STOP
   20 IF(NWAR) 70,30,50
C*** ABSOLUTE FIT
   30 DO 40 K = 1, NPT
   40 X(NWT, K) = 1.
      SUMWT=NPT
      C=1.
      GO TO 100
C*** RELATIVE FIT
   50 DO 60 K = 1, NPT
   60 X(NWT, K ) = 1./ X(NPCD, K ) **2
C*** WEIGHTS SUMMED
   70 SUMWT=X(NWT,1)
      DO 80 L = 2, NPT
   80 SUMWT = SUMWT + X(NWT, L)
      C=SQRT(SUMWT/FLOAT(NPT))
C*** CALCULATE QME OF FIT BEFORE (A(26,3)) AND AFTER (A(26,2)) ITERATING
  100 SUM2 = 0.
      DO 110 I=1,NPT
      CALL VF ( N    , X(1,I), DF, A ,D2F)
  110 SUM2 = SUM2 + X(NWT,I) * (X(9,I)-X(NPCD,I))**2
      A(26,2)=SQRT(SUM2/SUMWT)
      IF(N(4).EQ.0) GO TO 220
      IF(NIT.NE.0) GO TO 190
      NIT = N(4)
      A(26,3) = A(26,2)
C*** MAJOR ITER LOOP
      DO 180 K = 1, NIT
C*** SET UP SYSTEM OF LINEAR EQUATIONS..A(I,2)-V(I,J)*(A(J,1)-A°(J,1))=0
      DO 120 J = 1, NPAR
      A(J,2) = 0.
      DO 120 I = J,NPAR
      D2F(I,J)=0.
  120 V(I,J)=0.
      DO 150 KK=1,NPT
      CALL VF ( N    , X(1,KK), DF, A ,D2F)
      WDELY=(X(9,KK)-X(NPCD,KK))*X(NWT,KK)
```

```
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  310 PIVOT   =A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 352 L=1,N
      IF (PIVOT .NE. 0. )  GO TO 350
      A(ICOLUM,L)  = 0.
      GO TO 352
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
  352 CONTINUE
  355 IF(M) 380, 380, 360
  360 DO 375 L=1,M
      IF (PIVOT .NE. 0. )  GO TO 370
      B(ICOLUM,L)  = 0.
      GO TO 375
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
  375 CONTINUE
C
C     REDUCE NON-PIVOT ROWS
C
      BMAX=0.0
  380 DO 550 L1=1,N
  390 IF (L1 .EQ. ICOLUM) GO TO 550
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
      SUB =A(ICOLUM,L)*T
      A(L1,L)=A(L1,L)-SUB
C
C
      IF(INDEX(L1,3).EQ.1) GO TO 450
      IF(ABS(A(L1,L))-EPS4* ABS(SUB))449,449,450
  449 BMAX=AMAX1(BMAX,ABS(A(L1,L)))
  450 CONTINUE
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1) .EQ. INDEX(L,2)) GO TO 710
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
      DO 730 K = 1,N
```

```
      IF (INDEX(K,3) .NE. 1) GO TO 715
  730 CONTINUE
      RETURN
  715 ID=2
      RETURN
      END
```

```
      SUBROUTINE MATRIX(A,N1,I1,B,M1,DETERM,ID)
C
C*** MATRIX, WRITTEN BY CHARLES NEWMAN, IS AVAILABLE ON THE NSWC/WHITE OAK
C    PROGRAM LIBRARY.
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
C      TEST FOR LOST OF DIGITS DUE TO SUBTRACTION
C
      DIMENSION A(I1,I1),B(I1,M1),INDEX(50,3)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
      DATA EPS4/1.E-7/
C
C     INITIALIZATION
C
      BMAX=0.0
      ID=1
      M=M1
      N=N1
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 INDEX(J,3) = 0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
      IF (INDEX(J,3) .EQ. 1) GO TO 105
   60 DO 100 K=1,N
      IF(INDEX(K,3)-1) 80, 100, 100
   80 IF (AMAX .GE. ABS(A(J,K))) GO TO 100
   85 IROW=J
   90 ICOLUM=K
      AMAX =  ABS(A(J,K))
  100 CONTINUE
  105 CONTINUE
      INDEX(ICOLUM,3) = INDEX(ICOLUM,3) +1
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
      IF(BMAX.EQ.AMAX)ID=2
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW  .EQ. ICOLUM) GO TO 310
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
      IF(M) 310, 310, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
```

```
C*** COMPUTE A(I,2) VECTOR AND V MATRIX ELEMENTS
      DO 130 J=1,NPAR
      A(J,2)=A(J,2)-WDELY*DF(J)
      DO 130 I=J,NPAR
      V(I,J)=V(I,J)+X(NWT,KK)*DF(I)*DF(J)+WDELY*D2F(I,J)
  130 CONTINUE
  150 CONTINUE
      DO 160 J=1,NPARM1
      DO 160 I=J,NPARM1
  160 V(J,I+1)=V(I+1,J)
C*** INVERT MATRIX AND COMPUTE CHANGE IN °A° VECTOR
      CALL MATRIX ( V, NPAR, 28, A(1,2), 1, DETERM, ID)
C*** CALCULATE IMPROVED °A° VECTOR AND TEST FOR CONVERGENCE
C*** IF NO CONVERGENCE A(26,1)=0 ... IF CONVERGENCE A(26,1)=K
      A(26,1) = K
      DO 170 I = 1, NPAR
      M = A(I,3)
      A(M,1) = A(M,1) + A(I,2)
      IF (A(26,1) .EQ. 0.) GO TO 170
      SSS=ABS(A(I,2))
      IF(A(M,1).NE.0.) SSS=AMIN1(ABS(A(I,2)/A(M,1)),SSS)
      IF(SSS.GE.DELT) A(26,1)=0.
  170 CONTINUE
      IF (A(26,1) .NE. 0.) GO TO 100
  180 CONTINUE
      GO TO 100
C*** A(I,2) NOW BECOMES QME OF I-TH PARAMETER
  190 DO 210 I=1,NPAR
      M = A( I , 3 )
      IF(V(I,I) .GT. 0.) GO TO 200
      A(M,2)=-1.
      GO TO 210
  200 A(M,2) = A(26,2) * SQRT(V(I,I))
  210 CONTINUE
  220 RETURN
      END
```

```
      SUBROUTINE ORIENT
      DIMENSION WA(3,48),IWA(3,48)
      EQUIVALENCE (WA,IWA)
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      COMMON /BLOCK2/ AA(26,3),TITLE(8),XSHIFT,YSHIFT,ALPHA,BETA,
     1 CX(24,64),CY(24,64),MM,NN,NSEGS
      COMMON /BLOCK3/ PID180,YDCO(24),CA(28,24),RC2(64,24),X(9,64)
      LOGICAL SUBMERG,EQCAL,FLOODED
      REAL LBP
      IF(SUBMERG) 10,30
C
C*** VESSEL IS TOTALLY SUBMERGED.
C    CALCULATE TOTAL SECTION AREAS AND LOCATE THE SHIP ORIGIN AT THE
C    GEOMETRIC CENTERS.
   10 DO 20 IS=1,III
      II=IS
      JJ=JJA(II)
      KK=KKA(II)
      CALL AREACAL
      YCSO(II)=(DCU(II)+YCL(II))/2.
   20 CONTINUE
      RETURN
C
C*** VESSEL IS FLOATING AT SURFACE.
   30 READ 35,LBP,NSEGS,EQCAL
   35 FORMAT(E10.0,I10,L10)
C*   SET DEFAULT VALUE FOR NSEGS
      IF(NSEGS.EQ.0) NSEGS=20
      SEGWID=LBP/FLOAT(NSEGS)
      SEGDEN=SEGWID*DENSITY
      IF(EQCAL) GO TO 100
C
C*** SHIP EQUILIBRIUM POSITION IS NOT CALCULATED... INPUT WATERLINE IS USED.
C    CALCULATE ASSOCIATED BUOYANCY, FREEBOARD, DRAFT FOR EACH SECTION AND PRINT
      READ 40,WATERLN
   40 FORMAT(E10.0)
      BSUM=0.
      DO 50 I=1,III
      YCSO(I)=WATERLN-YDCO(I)
      JJ=JJA(I)
      KK=KKA(I)
      II=I
      CALL AREACAL
C*   WA AND IWA ARRAYS ARE USED FOR SORTING/PRINTING RESULTS.
      WA(1,I)=FLOAT(NS(I))*SEGWID
      WA(2,I)=AREA(I)*SEGDEN
      IWA(3,I)=I
      BSUM=BSUM+WA(2,I)
   50 CONTINUE
      CALL SORT(WA,III*3,3)
      PRINT 60,(TITLE(J),J=1,8)
   60 FORMAT(*1*4X,8A10//5X*SHIP EQUILIBRIUM POSITION NOT CALCULATED ...
     1 PROGRAM USES INPUT WATERLINE*)
```

```
      PRINT 65,WATERLN,BSUM
   65 FORMAT(*0*9X*INPUT WATERLINE (RELATIVE TO DATA ORIGIN) =*F6.2
     1/*0*9X*TOTAL BUOYANCY FORCE =*F10.2)
      PRINT 70
   70 FORMAT(*0*//14X*STATION*3X*FRAME*3X*FREEBOARD*3X*DRAFT*3X
     1*DIST FROM STA 0*3X*BUOYANCY*)
      DO 90 I=1,III
      K=IWA(3,I)
      J=JJA(K)
      DRAFT=YCSO(K)-YCL(K)
      FREEBD=RCIN(J,K)*SIN(PCIN(J,K))-YCSO(K)
      PRINT 80,NS(K),FRAME(K),FREEBD,DRAFT,WA(1,I),WA(2,I)
   80 FORMAT(16XI2,6XF5.1,5XF5.2,5XF5.2,7XF6.2,8XF8.2)
   90 CONTINUE
      GO TO 300
C
C*** FIND EQUILIBRIUM POSITION OF SHIP BY BALANCING SUMS OF WEIGHT AND
C    BUOYANCY FORCES (WSUM AND BSUM) AND MOMENTS (WMOM AND BMOM).
C    CALCULATE ASSOCIATED BUOYANCY, FREEBOARD, DRAFT FOR EACH SECTION AND PRINT
  100 LL=0
      READ 110, COLWID,WSCALE,ZSCALE,FLOODED
  110 FORMAT(3E10.0,L10)
      IF(COLWID.GT.1.01*SEGWID) PRINT 115
  115 FORMAT(*... WARNING WEIGHT HISTOGRAM COLUMN WIDTHS EXCEED THE SHIP
     1 SEGMENT LENGTHS.  ORIENT ASSUMES THE CONTRARY ...*)
C*   SET DEFAULT VALUE FOR WSCALE AND ZSCALE
      IF(ZSCALE.EQ.0.) ZSCALE=1.
      IF(WSCALE.EQ.0.) WSCALE=1.
C**  READ IN HISTOGRAM DISTANCE AND WEIGHT/LENGTH POINTS FROM DATA2 FILE.
C    DISTANCES ARE MEASURED FROM THE FORWARD PERPENDICULAR.
  120 READ(2,*) IDIG,ZDIG,WXDIG
      IF(EOF(2).NE.0) GO TO 125
      LL=LL+1
      WA(1,LL)=ZDIG*ZSCALE
      WA(2,LL)=WXDIG*WSCALE*COLWID
      IWA(3,LL)=-1
      GO TO 120
  125 NHIS=LL
C
C**  WSUM AND WMOM ARE CALCULATED FROM WEIGHT COLUMNS THAT OVERLAP SHIP
C    SEGMENTS OF CROSS SECTIONS APPEARING ON DATA1.  THIS RESULTS IN
C    MISSING SECTIONS BEING TREATED AS NEUTRAL.  HOWEVER, EXTRA WEIGHTS
C    DUE TO FLOODING, IF PRESENT, ARE INCLUDED BELOW IN WSUM AND WMOM.
C    ( PROGRAM ASSUMES COLWID .LE. SEGWID )
      DISPL=WSUMX=WSUM=WMOM=0.
      HLFCOL=COLWID/2.
      HLFSEG=SEGWID/2.
      DO 155 L=1,NHIS
      DISPL=DISPL+WA(2,L)
      ZWMIN=WA(1,L)-HLFCOL
      ZWMAX=WA(1,L)+HLFCOL
      ICHK=0
      DO 150 I=1,III
      ZB=NS(I)*SEGWID
      ZBMIN=ZB-HLFSEG
```

```
      ZBMAX=ZB+HLFSEG
      IF(ZWMIN.GE.ZBMAX.OR.ZWMAX.LE.ZBMIN) GO TO 150
      IF(ZWMIN.LT.ZBMIN) GO TO 130
      IF(ZWMAX.GT.ZBMAX) GO TO 140
      WSUM=WSUM+WA(2,L)
      WMOM=WMOM+WA(1,L)*WA(2,L)
      GO TO 155
  130 WLUMP=(ZWMAX-ZBMIN)/COLWID*WA(2,L)
      ZLUMP=(ZBMIN+ZWMAX)/2.
      GO TO 145
  140 WLUMP=(ZBMAX-ZWMIN)/COLWID*WA(2,L)
      ZLUMP=(ZWMIN+ZBMAX)/2.
  145 WSUM=WSUM+WLUMP
      WMOM=WMOM+WLUMP*ZLUMP
      IF(ICHK.EQ.1) GO TO 155
      ICHK=1
  150 CONTINUE
  155 CONTINUE
      IF(.NOT.FLOODED) GO TO 175
C
C*    READ IN EXTRA WEIGHT AND DISTANCE POINTS
      READ 160,NXTRA
  160 FORMAT(I10)
      DO 170 I=1,NXTRA
      LL=LL+1
      READ 165,ZXTRA,WXTRA
  165 FORMAT(2E10.0)
      WA(1,LL)=ZXTRA
      WA(2,LL)=WXTRA
      WMOM=WMOM+WXTRA*ZXTRA
      WSUMX=WSUMX+WXTRA
      IWA(3,LL)=0
  170 CONTINUE
      WSUM=WSUM+WSUMX
      DISPL=DISPL+WSUMX
C
C*** BEGIN ITERATION TO FIND SLOPE AND YINT FROM WHICH YCSO ARRAY IS CALCULATED.
C*    YCSO GIVES VERTICAL DISTANCE BETWEEN CALCULATION AND SHIP ORIGINS.
C
C*    ASSUME AN INITIAL DRAFT AND SLOPE FOR LINE CONNECTING SECTION WATERLINES.
C     POWER LAW IS FROM FIT TO TYPICAL SHIP DRAFT-DISPLACEMENT DATA.
  175 YINT=1.96*DISPL**0.26
      SLOPE=0.
C*    MODIFY YINT VALUE IF UNITS USED APPEAR TO BE METRIC.
      IF(DENSITY*35..GT.10.) YINT=YINT/3.
C
  180 BSUM=BMOM=BP=BPZ=BPZ2=0.
      DO 190 IS=1,III
      II=IS
      ZBI=FLOAT(NS(II))*SEGWID
      YCSO(II)=YINT+SLOPE*ZBI-YDCO(II)
      JJ=JJA(II)
      KK=KKA(II)
      CALL AREACAL
      BI=AREA(II)*SEGDEN
```

```
      BPI=2.*(DCU(II)-XCSO(II))*SEGDEN
      BSUM=BSUM+BI
      BMOM=BMOM+BI*ZBI
      BP=BP+BPI
      BPZ=BPZ+BPI*ZBI
      BPZ2=BPZ2+BPI*ZBI**2
      WA(1,LL+II)=ZBI
      WA(2,LL+II)=BI
      IWA(3,LL+II)=II
  190 CONTINUE
      D1=WSUM-BSUM
      D2=WMOM-BMOM
      IF(ABS(D1/WSUM)+ABS(D2/WMOM).LT.1.E-9) GO TO 200
      DTERM=(BP*BPZ2-BPZ**2)
      YINT=YINT+(D1*BPZ2-D2*BPZ)/DTERM
      SLOPE=SLOPE+(D2*BP-D1*BPZ)/DTERM
      GO TO 180
C
C*** PRINT EQUILIBRIUM CALCULATION RESULTS.
  200 AKEEL=ATAN(SLOPE)/PID180
      NWA=LL+III
      CALL SORT(WA,NWA*3,3)
      PRINT 210,(TITLE(J),J=1,8)
  210 FORMAT(*1*4X,8A10//5X*RESULTS OF SHIP EQUILIBRIUM CALCULATION ...
     1(UNLISTED STATIONS CONSIDERED NEUTRAL)*)
      PRINT 215,WSUMX
  215 FORMAT(*0*9X*EXTRA WEIGHT DUE TO FLOODING =*F8.2)
      PRINT 220,DISPL,AKEEL
  220 FORMAT(*0*9X*DISPLACEMENT =*F9.2//10X*ANGLE OF KEEL = *F5.2
     1* DEGREES FROM HORIZONTAL  (POS = BOW UP)*)
      PRINT 230
  230 FORMAT(*0*//14X*STATION*3X*FRAME*3X*FREEBOARD*3X*DRAFT*3X
     1*DIST FROM STA 0*3X*BUOYANCY*4X*W(DATA2)*)
      IF(NXTRA.GT.0) PRINT 235
  235 FORMAT(*+*T95*W(EXTRA)*)
      DO 250 I=1,NWA
      K=IWA(3,I)
      IF(K.EQ.-1) PRINT 240,WA(1,I),WA(2,I)
  240 FORMAT(T57,F6.2,T82,F8.2)
      IF(K.EQ.0 ) PRINT 245,WA(1,I),WA(2,I)
  245 FORMAT(T57,F6.2,T95,F7.2)
      IF(K.LE.0) GO TO 250
      J=JJA(K)
      FREEBD=RCIN(J,K)*SIN(PCIN(J,K))-YCSO(K)
      DRAFT=YCSO(K)-YCL(K)
      PRINT 80,NS(K),FRAME(K),FREEBD,DRAFT,WA(1,I),WA(2,I)
  250 CONTINUE
  300 RETURN
      END
```

```
      SUBROUTINE SHIPLOT(MODE)

** SHIPLOT IS A GOULD OR CALCOMP PLOTTING SUBROUTINE. MOST OF THE
   SUBROUTINE CALLS DEPEND ON A PROPRIETARY SOFTWARE PACKAGE FOR
   A GOULD ELECTROSTATIC PLOTTER.

   SECTIONS ARE PLOTTED ON INDIVIDUAL PLOTS (MULTI=TRUE)
   OR TOGETHER ON A SINGLE PLOT (MULTI=FALSE).
   CONTENT OF EACH PLOT IS DETERMINED BY FLAGS SET IN THE IPLOT ARRAY.
   IPLOT(I)=0 ...SKIP I-TH PLOT CATEGORY.
   IPLOT(1)=N ...DRAW FITTED SPLINE FUNCTION CURVE. (FOR N SEE BELOW)
   IPLOT(2)=N ...PLOT INPUT POINTS DEFINING SECTION.
   IPLOT(3)=N ...DRAW FITTED MAPPING FUNCTION CURVE.
   IPLOT(4)=N ...PLOT Z-PLANE IMAGES OF POINTS ON UNIT CIRCLE.

   N IS AN INTEGER FROM 1 TO 9 THAT CONTROLS THE THICKNESS OF THE PLOTTED
   POINT OR LINE.  A VALUE OF 4 PRODUCES A STANDARD THICKNESS.  SMALLER
   VALUES ARE LIGHTER AND LARGER VALUES ARE DARKER.

   FMAG IS A MAGNIFICATION FACTOR THAT AFFECTS ALL PLOT DIMENSIONS (1.=1 TO 1)
   XEND AND YEND ARE LENGTHS OF THE X AND Y AXES RELATIVE TO THE SHIP ORIGIN.
   THESE MUST BE GIVEN IN TENS OF FEET (E.G.,10,20,30,40).

** ALL SHIPFIT CALLS TO PLOTTING SUBROUTINES ARE CONTAINED IN SHIPLOT.
** IT IS CALLED BY SHIPFIT WITH THE FOLLOWING ARGUMENT VALUES,
   MODE=1 ...INITIALIZE PLOTS.
   MODE=2 ...PLOT II-TH SECTION (ACCORDING TO IPLOT VARIABLE).
   MODE=3 ...END ALL PLOTTING.

      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      COMMON /BLOCK2/ AA(26,3),TITLE(8),XSHIFT,YSHIFT,ALPHA,BETA,
     1 CX(24,64),CY(24,64),MM,NN,NSEGS
      COMMON /BLOCK3/ PID180,YDCO(24),CA(28,24),RC2(64,24),X(9,64)
      DIMENSION XPLT(200),YPLT(200),IPLOT(10),PIN(9)
      LOGICAL SUBMERG,MULTI
      DATA PIN/.125,.25,.5,1.,2.,3.,4.,6.,9./
      DATA BCDX,BCDY/1HX,1HY/
      DATA PID2/1.5707963267949/

      IF(MODE.EQ.1) GO TO 1
      IF(MODE.EQ.2) GO TO 35

** END ALL PLOTTING.
      CALL PEND
      GO TO 100

** INITIALIZE PLOT PARAMETERS AND DRAW AXES IF SINGLE PLOT.

   READ (AND SET DEFAULT VALUES FOR) FMAG, XEND, AND YEND.
    1 READ  5,(IPLOT(I),I=1,10),MULTI,FMAG,XEND,YEND
    5 FORMAT(10I1,L10,3E10.0)
      IF(FMAG.EQ.0.) FMAG=1.
      IF(XEND.EQ.0.) XEND=30.
```

```
      IF(YEND.EQ.0.) YEND=30.
C
      NXDIV=XEND/10.+.5
      NYDIV=YEND/10.+.5
      NPLT=198
      XPLT(199)=YPLT(199)=0.
      THSTP=PID2/FLOAT(NPLT-1)
      IF(MULTI) GO TO 20
C
C**   SET PARAMETERS FOR SINGLE PLOT AND DRAW AXES.
      CALL PSIZE(-1.,-0.5,16.,10.,-1)
      CALL FACTOR(FMAG*2.,FMAG*2.)
      IFLIP=III/2
      IF(SUBMERG) GO TO 10
C
C*    SET LENGTH OF NEGATIVE Y AXIS (REL.TO SO) FOR SURFACE SHIP.
      YL=6.
      ISO=1
      YOY=2.
      GO TO 15
C
C*    SET LENGTH OF NEGATIVE Y AXIS (REL.TO SO) FOR SUBMERSIBLE.
   10 YL=3.
      ISO=2
      YOY=2.
      THSTP=2.*THSTP
   15 YOX=YOY+YL*ISO
      TENFT=YL/NYDIV
      XL=NXDIV*TENFT
      XPLT(200)=XEND/XL
      YPLT(200)=YEND/YL
C
C*    DRAW X AND Y AXES FOR SINGLE PLOT.
      CALL INTENS(2.)
      CALL MOVEA(0.,YOX)
      CALL ANGLE(0.)
      CALL AXCCW
      CALL AXDEF(2*NXDIV*TENFT,2*NXDIV)
      CALL TXTSIZ(.1,.05,.1)
      CALL TXTFMT(°F°,3,0)
      CALL AXVAL(-XEND,10.)
      CALL MOVEA(XL,YOY)
      CALL YAXIS(YL*ISO,ISO*NYDIV)
      CALL SYMBOL(XL-3.,YOY-.5,.14,TITLE,0.,80)
      CALL REORGA(XL,YOY+YL)
      GO TO 100
C
C**   SET PARAMETERS FOR MULTIPLE PLOTS.
   20 IF(SUBMERG) GO TO 25
C
C*    SET LENGTH OF NEGATIVE Y AXIS (REL.TO SO) FOR SURFACE SHIP.
      YL=6.
      ISO=1
      YOY=2.
      GO TO 30
```

```
C
C*   SET LENGTH OF NEGATIVE Y AXIS (REL.TO SO) FOR SUBMERSIBLE.
   25 YL=3.
      ISO=2
      YOY=2.
      THSTP=2.*THSTP
   30 YOX=YOY+YL*ISO
      TENFT=YL/NYDIV
      XL=TENFT*NXDIV
      XPLT(200)=XEND/XL
      YPLT(200)=YEND/YL
      GO TO 100
C
C
C*** CREATE AND PLOT DATA ARRAYS (DRAW AXES IF MULTIPLE PLOTS).
   35 IF(.NOT.MULTI) GO TO 40
C
C**  DRAW AND LABEL AXES FOR II-TH SECTION.
      CALL PSIZE(-1.,-0.5,16.,10.,-1)
      CALL FACTOR(FMAG*2.,FMAG*2.)
      CALL INTENS(2.)
      CALL MOVEA(0.,YOX)
      CALL ANGLE(0.)
      CALL AXCCW
      CALL AXDEF(XL,NXDIV)
      CALL TXTSIZ(.1,.05,.1)
      CALL TXTFMT(°F°,3,0)
      CALL AXVAL(0.,10.)
      CALL TXTSIZ(.15,.1,.15)
      CALL AXLAB(1,BCDX)
      CALL MOVEA(0.,YOY)
      CALL YAXIS(YL*ISO,ISO*NYDIV)
      CALL TXTSIZ(.1,.05,.1)
      CALL TXTFMT(°F°,3,0)
      CALL AXVAL(-YEND,10.)
      CALL TXTSIZ(.15,.1,.15)
      CALL AXLAB(1,BCDY)
      CALL SYMBOL(0.,YOY-.5, .14,TITLE,0.,70)
      CALL SYMBOL(XL-2.5,YOY+.5,.14,14HSTATION NUMBER,0.,14)
      CALL NUMBER(XL-.4,YOY+.5,.14,FLOAT(NS(II)),0.,0)
      CALL REORGA(0.,YOY+YL)
C
C**  PLOT DATA FOR II-TH SECTION.
   40 XSIGN=1.
      IF(.NOT.MULTI.AND.II.GT.IFLIP) XSIGN=-1.
      IF(IPLOT(1).EQ.0) GO TO 60
C
C*   DRAW SPLINE FUNCTION CURVE (FITTED TO INPUT POINTS).
      IF(SUBMERG) GO TO 45
      PL=ATAN2(YCL(II),XCSO(II))
      PU=ATAN2(YCSO(II),DCU(II))
      GO TO 50
   45 PL=ATAN2(YCL(II),XCSO(II))
      PU=ATAN2(DCU(II),XCSO(II))
   50 PRANGE=PU-PL
```

```
      PSTP=PRANGE/FLOAT(NPLT-1)
      P=PL-PSTP
      DO 55 I=1,NPLT
      P=P+PSTP
      CALL SPLCAL
      C=COS(P)
      S=SIGN(SQRT(1.-C*C),P)
      XPLT(I)=(R*C-XCSO(II))*XSIGN
      YPLT(I)=(R*S-YCSO(II))
   55 CONTINUE
      CALL INTENS(PIN(IPLOT(1)))
      CALL LINE(XPLT,YPLT,NPLT,1,0,0)
   60 IF(IPLOT(2).EQ.0) GO TO 75
C
C*    PLOT INPUT POINTS DEFINING II-TH SECTION.
      DO 65 J=1,JJ
      NJS=J
      C=COS(PCIN(J,II))
      S=SIGN(SQRT(1.-C*C),PCIN(J,II))
      XPLT(J)=(RCIN(J,II)*C-XCSO(II))*XSIGN
      YPLT(J)=(RCIN(J,II)*S-YCSO(II))
      IF(SUBMERG) GO TO 65
      IF(YPLT(J).LE.0.) GO TO 65
      NJS=J-1
      GO TO 70
   65 CONTINUE
   70 XPLT(NJS+1)=YPLT(NJS+1)=0.
      XPLT(NJS+2)=XEND/XL
      YPLT(NJS+2)=YEND/YL
      CALL INTENS(PIN(IPLOT(2)))
      CALL LINE(XPLT,YPLT,NJS,1,-1,1)
   75 IF(IPLOT(3).EQ.0) GO TO 92
C
C*    DRAW MAPPING FUNCTION CURVE FOR II-TH SECTION.
      DO 80 M=1,NPLT
      XPLT(M)=0.
      YPLT(M)=0.
   80 CONTINUE
      DO 90 M=1,NPLT
      THETAM=THSTP*FLOAT(M-1)-PID2
      DO 90 N=1,NN
      ARG=(ALPHA-BETA*N)*THETAM
      IF(SUBMERG.AND.MOD(N,2).EQ.0) GO TO 85
      XPLT(M)=XPLT(M)+AA(N,1)*COS(ARG)
      YPLT(M)=YPLT(M)+AA(N,1)*SIN(ARG)
      GO TO 90
   85 XPLT(M)=XPLT(M)-AA(N,1)*SIN(ARG)
      YPLT(M)=YPLT(M)+AA(N,1)*COS(ARG)
   90 CONTINUE
      CALL INTENS(PIN(IPLOT(3)))
      CALL LINE(XPLT,YPLT,NPLT,1,0,0)
   92 IF(IPLOT(4).EQ.0) GO TO 100
C
C*    PLOT MAPPING FUNCTION IMAGE POINTS FOR THE II-TH SECTION.
      DO 95 M=1,MM
```

```
      XPLT(M)=X(6,M)
      YPLT(M)=X(7,M)
   95 CONTINUE
      XPLT(MM+1)=YPLT(MM+1)=0.
      XPLT(MM+2)=XEND/XL
      YPLT(MM+2)=YEND/YL
      CALL  INTENS(PIN(IPLOT(4)))
      CALL LINE(XPLT,YPLT,MM,1,-1,2)
  100 RETURN
      END
```

```
      SUBROUTINE SORT(A,NA,LG)
C
C     THIS IS A SIMPLE INSERTION SORT
C
      DIMENSION A(LG,1),T(25)
C
      NG=NA/LG
      IF(NG .LT. 2) RETURN
C
      DO 150 I = 2,NG
      IF(A(1,I).GE.A(1,I-1)) GO TO 150
      J=I-1
   50 DO 60 K=1,LG
      T(K)=A(K,I)
      A(K,J+1)=A(K,J)
   60 CONTINUE
      J=J-1
C     THE TEST FOR (J .EQ. 0) IS FASTER THAN A TEST FOR (J .EQ. 1)
      IF(J .EQ. 0) GO TO 100
      IF(T(1) .LT. A(1,J)) GO TO 50
C     A SUBSCRIPT OF THE FORM (0+1) IS ACCEPTABLE.
  100 DO 110 K=1,LG
      A(K,J+1)=T(K)
  110 CONTINUE
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPLCAL
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      IJ=KK-1
      SUM1=SUM2=SUM3=0.
      DO 2    K=1,JJ,IJ
      IF(P.LE.PCIN(K,II)) GO TO 3
2     CONTINUE
      K=JJ
3     IF(K.LE.KK) GO TO 5
      K=(K-KK)/IJ
      DO 4    J=1,K
      DP=PCIN(J*IJ+1,II)-P
      C341=(CF(J+3,II)-CF(J+4,II))*DP
      C342=C341*DP
      C343=C342*DP
      SUM1=SUM1+C341
      SUM2=SUM2+C342
4     SUM3=SUM3+C343
5     RPP=(2.*CF(3,II))+6.*((P*CF(4,II))+SUM1)
      RP=CF(2,II)+P*((2.*CF(3,II))+3.*(P*CF(4,II)))-3.*SUM2
      R=CF(1,II)+P*(CF(2,II)+P*(CF(3,II)+(P*CF(4,II))))+SUM3
      RETURN
      END
```

```
      SUBROUTINE SPLINT (N,X,Y,JJ,A,R,S,C)
      DIMENSION X(1),Y(1),C(25)
      IJ=JJ-1
   13 SUM=0.
      DO 500 K=1,N,IJ
      IF (S-X(K)) 400,400,500
  500 CONTINUE
      K=N
  400 IF(K-JJ) 200,200,600
  600 K=(K-JJ)/IJ
  300 DO 800 J=1,K
      IK=J*IJ+1
  800 SUM=SUM+(C(J+3)-C(J+4))*(X(IK)-S)**4
  200 SUM1=0.
      DO 501 K=1,N,IJ
      IF (R-X(K)) 401,401,501
  501 CONTINUE
      K=N
  401 IF(K-JJ) 201,201,601
  601 K=(K-JJ)/IJ
  301 DO 801 J=1,K
      IK=J*IJ+1
  801 SUM1=SUM1+(C(J+3)-C(J+4))*(X(IK)-R)**4
  201 A=(C(1)+C(2)*S/2.+C(3)*S**2/3.+C(4)*S**3/4.)*S-(C(1)+C(2)*R/2.
     1+C(3)*R**2/3.+C(4)*R**3/4.)*R-.25*(SUM-SUM1)
   11 RETURN
       END
```

```
      SUBROUTINE SPLSQ1 (NPT,X,Y,C,E,JJ)
C
C*** SPLSQ1 FITS A CUBIC SPLINE FUNCTION TO INPUT DATA BY A LEAST SQUARES
C    TECHNIQUE.  IT WAS WRITTEN BY CHARLES WEBER AND IS AVAILABLE THROUGH
C    THE NSWC/WHITE OAK PROGRAM LIBRARY.
C
      DIMENSION DF(25),X(1),Y(1),C(1),V(25,25)
      IJ=JJ-1
      NR=(NPT-1)/IJ+3
      NUFSED=0
      DO 1 I=1,NR
      C(I)=0.
      DO 1 J=1,NR
    1 V(I,J)=0.
    8 DO 10 I=1,NR
   10 DF(I)=0.
      DO 20 I=1,NPT
      DF(1)=1.
      DF(2)=X(I)
      DF(3)=X(I)**2
      DF(4)=DF(3)*X(I)
      IF(I.LE.JJ) GO TO 6
    4 DF(4)=DF(4)+(X(JJ)-X(I))**3
      N=I-(I*(JJ-2)+1)/IJ
      IF(N.GT.NR-3) N=NR-3
      IK=IJ*(N-1)+1
      DF(N+3)=-(X(IK)-X(I))**3
      IF(N.EQ.2) GO TO 6
      N2=N+2
      DO 5 K=5,N2
      IK=(K-4)*IJ+1
      IL=IK+IJ
    5 DF(K)=-(X(IK)-X(I))**3+(X(IL)-X(I))**3
    6 IF(NUFSED.EQ.0) GO TO 7
      Y1=0.
      DO 3 J=1,NR
    3 Y1=Y1+C(J)*DF(J)
      SUM=SUM+(Y1-Y(I))**2
      GO TO 20
    7 DO 2 J=1,NR
      C(J)=C(J)+Y(I)*DF(J)
      DO 2 K=1,NR
    2 V(K,J)=V(K,J)+DF(K)*DF(J)
   20 CONTINUE
      IF(NUFSED.EQ.1) GO TO 9
      CALL MATRIX (V,NR,25,C,1,DE,ID)
      SUM=0.
      NUFSED=1
      IF(E.LT.0.) RETURN
      GO TO 8
    9 E=SQRT(SUM/FLOAT(NPT-NR))
      RETURN
       END
```

```
      SUBROUTINE SUBBEAM(BD2)
C*** SUBBEAM CALCULATES THE HALF BEAM OF A SUBMERSIBLE FORM
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      XCMAX=0.
      DO 10 J=1,JJ
      XC=RCIN(J,II)*COS(PCIN(J,II))
      XCMAX=AMAX1(XCMAX,XC)
      IF(XCMAX.EQ.XC) JMAX=J
   10 CONTINUE
      P=PCIN(JMAX,II)
   20 CALL SPLCAL
      C=COS(P)
      T=TAN(P)
      F=RP-R*T
      IF(F.EQ.0.) GO TO 30
      DELP=F/(RPP-RP*T-R/C**2)
      P=P-DELP
      IF(DELP/P.LT.1.E-6) 30,20
   30 BD2=R*C-XCSO(II)
      RETURN
      END
```

```
      SUBROUTINE XLIMIT(PIN,XC)
C*** XLIMIT FINDS BY ITERATION THE INTERSECTION OF THE SHIP FORM CURVE AND THE
C    ABSCISSA OF THE SHIP COORDINATE SYSTEM. OUTPUT IS RELATIVE TO CO.
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      P=PIN
   10 CALL SPLCAL
      S=SIN(P)
      C=SQRT(1.-S*S)
      F=R*S-YCSO(II)
      IF(F.EQ.0.) GO TO 20
      DELP=F/(RP*S+R*C)
      P=P-DELP
      IF(ABS(DELP/P).LT.1.E-6) 20,10
   20 XC=R*C
      RETURN
      END

      SUBROUTINE YLIMIT(PIN,YC)
C*** YLIMIT FINDS BY ITERATION THE INTERSECTION OF THE SHIP FORM CURVE AND THE
C    ORDINATE OF THE SHIP COORDINATE SYSTEM IN THE VICINITY OF THE CO POINT
C    (PIN,R(PIN)). OUTPUT IS RELATIVE TO CO.
      COMMON /BLOCK1/ SUBMERG,NS(24),FRAME(24),KKA(24),KK,JJA(24),JJ,
     1 XCSO(24),YCSO(24),RCIN(64,24),PCIN(64,24),CF(28,24),E(24),DCU(24)
     2,YCL(24),AREA(24),DENSITY,P,R,RP,RPP,II,III
      P=PIN
   10 CALL SPLCAL
      C=COS(P)
      S=SIGN(SQRT(1.-C*C),P)
      F=R*C-XCSO(II)
      IF(F.EQ.0.) GO TO 20
      DELP=F/(RP*C-R*S)
      P=P-DELP
      IF(ABS(DELP/P).LT.1.E-6) 20,10
   20 YC=R*S
      RETURN
      END
```

DISTRIBUTION

	Copies
Chief of Naval Material	
Attn: A. Bernard (PM-4)	1
Dr. A. J. Faulstich (MAT-07)	1
Navy Department	
Washington, DC 20360	
Commander	
Naval Sea Systems Command	
Attn: SEA-63R (F. Romano)	1
SEA-55X (R. Keane)	1
SEA-55X (J. Schell)	1
SEA-55X (S. Herman)	1
SEA-55X (J. Fowler)	1
SEA-55X (D. Hurt)	1
PMS-406 (R. Whitman)	1
Department of the Navy	
Washington, DC 20362	
Commander	
David W. Taylor Naval Ship Research and Development Center	
Attn: Code 17 (Dr. W. Murray)	1
Code 1730.2 (D. Harry)	1
Code 1740 (Dr. F. Fisch)	1
Code 175 (J. Sykes)	1
Code 175.3 (W. Conley)	1
Code 175.3 (P. Manny)	1
Code 175.3 (Dr. B. Wang)	1
Bethesda, MD 20084	
David W. Taylor Naval Ship Research and Development Center	
Attn: Code 177 (R. Fuss)	1
Code 177.1 (V. Bloodgood)	1
Code 177.1 (M. Riley)	1
Code 177.1 (R. Higginbotham)	1
Portsmouth, VA 23709	

	Copies
Commander	
Naval Weapons Center	
Attn: Code 533 (Technical Library)	1
China Lake, CA 93555	
Commander	
Naval Ocean Systems Center	
Attn: Technical Library	1
San Diego, CA 92152	
Commander	
Naval Underwater Systems Center	
Attn: Technical Library	1
Newport, RI 02840	
Office of Naval Research	
Attn: Code 474 (Dr. A. Kushner)	1
800 North Quincy Street	
Arlington, VA 22217	
Defense Nuclear Agency	
Attn: SPSS (CDR J. White)	1
Washington, DC 20305	
Defense Technical Information Center	
Cameron Station	
Alexandria, VA 22314	12
Library of Congress	
Attn: Gift and Exchange Division	4
Washington, DC 20540	
Internal Distribution:	
G402 (Dr. P. Huang)	1
R	1
R10	1
R102 (D. Phillips)	1

DISTRIBUTION (Cont.)

	Copies
Internal Distribution (Cont.)	
R102 (K. W. Reed)	1
R14	1
R14 (Dr. K. Bannister)	1
R14 (R. Bendt)	1
R14 (T. Farley)	1
R14 (J. F. Goertner)	1
R14 (Dr. F. Hains)	1
R14 (G. Harris)	1
R14 (A. Henney)	1
R14 (N. Holland)	1
R14 (Dr. H. Huang)	1
R14 (K. Kiddy)	1
R14 (J. Koenig)	1
R14 (M. Marcus)	1
R14 (M. Marshall)	1
R14 (W. McDonald)	25
R14 (Dr. M. Moussouros)	1
R14 (R. Thun)	1
R14 (S. Wilkerson)	1
R15	1
R121 (M. Stosz)	1
R122 (L. Roslund)	1
R44 (Dr. A. Van Tuyl)	1
E35 (GIDEP Office)	1
E231	9
E232	3

END

FILMED

9-85

DTIC